บทความพิเศษ

กราฟิก บนไมโครคอมพิวเตอร์

"จุดใหญ่จะพูดถึงระบบคอมพิวเตอร์กราฟิกที่อยู่บนเครื่อง IBM PC เพื่อเป็นความรู้ในการพิจารณาจัดซื้อให้เหมาะสมกับงานของท่านเอง"

ดร.ไพศาล สงวนหมู่

กล่าวนำ

ผมติดตามข่าวคราวของเทคโนโลยีทางด้านอิเล็กทรอนิกส์และคอมพิวเตอร์ทางวารสารและข่าวคราวทางหนังสือพิมพ์เป็นประจำ ทุกครั้งที่ผมนำหนังสือเหล่านี้ขึ้นมาอ่านมีความรู้สึกว่า ตัวเองยังไม่อยากจะตายไปง่าย ๆ กิเลสในใจเรียกร้องให้ผมติดตามดูต่อไปว่าอีก 40-50 ปีข้างหน้า จะมีอะไรใหม่ ๆ ที่มนุษย์จะสามารถพัฒนาหรือคิดค้นขึ้นมาจะมีหน้าตาเป็นอย่างไรบ้าง เห็นอะไรใหม่ ๆ ในหน้าหนังสือพิมพ์หรือในรายงานก็ให้เกิดมีความต้องการอยากจะจับอยากจะเล่นไปเสียหมด ผมจำได้ว่าสมัยเรียนคอมพิวเตอร์ใหม่ ๆ กับอาจารย์สวัสดิ์ แสงบางปลา (ขออนุญาตเอ่ยนามท่านเสียในที่นี้ในฐานะที่ท่านเป็นปรมาจารย์ของผม) ความรู้สึกมันไม่รุนแรงเหมือนกับตอนนี้ อาจจะเป็นเพราะว่าตอนนั้นจะทำโปรแกรมคอมพิวเตอร์ทีก็ต้องไปนั่งเสียเวลาเจาะบัตร เสียเวลาเฝ้าดูผลที่จะออกมาทางเครื่องพิมพ์อย่างน้อย 2-3 วัน แล้วก็เสียเวลาไปแก้โปรแกรมใหม่ รวมแล้วทั้งหมดอาจจะทำได้โปรแกรมเดียวใน 1 เทอม ไปเรียนต่อเมืองนอกก็ดีขึ้นมาหน่อย เสียเวลารอผลการทำงานแค่ 2-3 ชั่วโมง สมัยนี้นะเหรอ แสนสบาย นั่งทำโปรแกรมส่งอาจารย์ที่บ้านยังได้ ผมว่าเทคโนโลยีทางด้านนี้มันไปเร็วกว่า function exponential ที่ใช้แบบสัญลักษณ์ของมหาวิทยาลัยศรีนครินทรวิโรฒเสียอีก

ผมว่าท่านผู้อ่านหลายท่านคงมีความรู้สึกเช่นเดียวกับผมในเรื่องของความอยากรู้อยากเห็นอยากลองในเทคโนโลยีใหม่ ๆ ไม่เช่นนั้นผมว่า คนคิดเทคโนโลยีใหม่ ๆ คงจะขายไม่ค่อยออก สมัยที่บริษัท Apple ออกไมโครคอมพิวเตอร์รุ่นแรกออกมา ขายดีแบบเทน้ำเทท่า ผมก็อยากได้แต่ไม่มีปัญญาซื้อในสมัยที่ขับเคี่ยวกับอาจารย์ในเรื่องวิทยานิพนธ์ ไม่ทันที่เจ้าเครื่อง Apple จะถูกนำมาใช้งานในด้านธุรกิจเท่าไหร่เลย IBM ก็ออก PC มา Apple รุ่นแรกก็ตกกระป๋อง ถามกันจริง ๆ เถอะมีใครทันได้ใช้ Apple ในทางธุรกิจอย่างจริงจังกันบ้าง พอถึงยุคของ IBM/PC การใช้งานเริ่มจะกว้างขึ้น การเปลี่ยนแปลงก็ยังคงดำเนินต่อไป

ในรูปแบบของการเปลี่ยนแปลงนั้น ขีดความสามารถทางด้านกราฟิกก็ถูกพัฒนาขึ้นมาตามขีดความสามารถของซีพียู เราเริ่มเห็นภาพกราฟิกที่มีความละเอียดมากขึ้น บนเครื่องพีซี VGA ดูเหมือนจะเป็นมาตรฐานสำหรับเครื่องพีซีไปเสียแล้ว ถึงเวลานี้เมื่อฮาร์ดแวร์มีขีดความสามารถในการแสดงกราฟิกมากขึ้น ซอฟต์แวร์ก็พัฒนาตามมาทำให้ผู้ใช้งานสามารถใช้เครื่องได้อย่าง Friendly หรือสนิทสนมมากขึ้น Graphical User Interface (GUI) เริ่มมีการวางรากฐานของมาตรฐานเพื่อให้ผู้ใช้งานใช้รูปสัญลักษณ์ (ICON) แทนคำสั่งต่าง ๆ บนหน้าจอแทน

คงจำกันได้ว่า ไมโครฯ ฉบับที่ 43 ประจำเดือนตุลาคม 31 ก็เคยนำเรื่องคอมพิวเตอร์กราฟิกมานำเสนอไปบ้างแล้ว

ในส่วนของผมในฉบับนี้จะเป็นการเล่าสู่กันฟังเสียมากกว่าที่จะกล่าวถึงทางด้านวิชาการมากเกินไป จุดใหญ่ก็จะพูดถึงระบบคอมพิวเตอร์กราฟิกที่มีอยู่บน IBM PC คอมแพตติเบิ้ลในปัจจุบันเพื่อทำความเข้าใจเสียก่อนว่า มีอุปกรณ์อะไรอยู่เบื้องหน้าเบื้องหลังบ้าง เผื่อว่าจะได้เป็นความรู้ในการพิจารณาจัดซื้อให้เหมาะสมกับงานของท่านเอง เพราะผู้ขายส่วนมากก็จะพูดถึงส่วนดี ๆ ของสินค้า จนผู้ซื้อนึกจินตนาการไปไกลกว่าความเป็นจริง พอซื้อมาก็รู้สึกผิดหวัง นอกเหนือจากนั้นก็จะเสริมความรู้ด้านเทคนิคลงไปบ้าง ผสมผสานกันจนได้รสชาติที่พอเหมาะไม่เลี่ยนวิชาการจนน่าเบื่อ

คอมพิวเตอร์กราฟิกกับการนำไปใช้งาน

คำพูดของฝรั่งที่ว่า "A pictures worth thousand words" หรือภาพใช้แทนคำพูดได้เป็นพัน ๆ คำ เห็นจะไม่ผิด

ไปจากความเป็นจริงเท่าไหร่นัก เพราะรูปภาพดูได้เข้าใจง่ายกว่าการอ่านคำบรรยายมากมายนัก หนังสือเล่มไหนไม่มีรูปภาพประกอบดูเหมือนจะสร้างกำแพงความเข้าใจระหว่างผู้เขียนกับผู้อ่านเสียตั้งแต่แรก

การทำให้คอมพิวเตอร์แสดงภาพจึงเป็นสิ่งที่จะเสริมคุณค่าของคอมพิวเตอร์ให้มากขึ้น การพัฒนาทางด้านกราฟิกบนไมโครคอมพิวเตอร์จึงดำเนินมาตลอดจนถึงขั้นที่ผู้ใช้พอใจในราคาและคุณภาพในระดับหนึ่ง

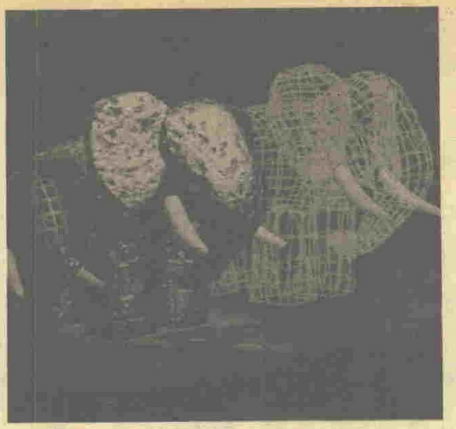
คอมพิวเตอร์กราฟิกที่ใช้กันในคอมพิวเตอร์มีหลายหลากชนิด แต่ก็พอจะแบ่งประเภทออกได้เป็น 7 ประเภทใหญ่ ๆ คือ

1. โปรแกรมสำหรับวาดภาพ (Paint Programs)

โปรแกรมวาดภาพเป็นการสเก็ตภาพอย่างหยาบ ๆ โดยการใช้มือหรือการนำภาพที่ได้จากเครื่องรับภาพเข้าสู่คอมพิวเตอร์ (เช่น scanner) มาเปลี่ยนแปลงแก้ไขสำหรับงานพิมพ์ภาพที่ได้จากโปรแกรมนี้จะอยู่ในรูปของบิตแม็พ ซึ่งจะมีคุณภาพด้อยลงไปเมื่อถูกนำมาขยาย และคุณภาพของภาพจะขึ้นอยู่กับฮาร์ดแวร์ที่ใช้ โปรแกรมพวกนี้จะใช้รูปแบบไฟล์เก็บแฟ้มที่มีนามสกุล .PCX ตัวอย่างเช่น PC Paintbrush, Publisher, Window Paint หรือ Drhalo เป็นต้น

2. โปรแกรมสร้างภาพประกอบและการออกแบบ (Illustration /Design Software)

โปรแกรมประเภทนี้สร้างภาพในลักษณะเวกเตอร์ ใช้การลากเส้นตรงเส้นโค้งเป็นหลัก มีรูปแบบการสร้างสีสันที่ค่อนข้างจะยุ่งยาก ความนิยมของโปรแกรมประเภทนี้พอ ๆ กับโปรแกรมวาดภาพ เนื่องจากคุณภาพของภาพที่ได้ออกมาไม่ได้ผูกติดกับฮาร์ดแวร์ตัวที่สร้างภาพนั้น คือถ้าอุปกรณ์



เขียนภาพที่มีความละเอียดสูงก็จะได้ภาพที่มีความละเอียดสูง ภาพที่สร้างโดยโปรแกรมประเภทนี้ส่วนมากอยู่ในรูปของ CGM (Computer Graphic Matafile) ซึ่งเป็นรูปแบบที่เตรียมตัวจะเป็นรูปแบบมาตรฐานของแฟ้ม สำหรับภาพที่อยู่ในรูปของเวกเตอร์ ตัวอย่างของโปรแกรมประเภทนี้ก็เช่น Adobe Illustrator, Corel Draw, GEM Artline, MASS-11 Draw และ Micrografix Designer

3. โปรแกรมกราฟิกเพื่อการนำเสนอ (Presentation Graphic Program)

โปรแกรมกราฟิกสำหรับการนำเสนอช่วยทำตัวเลขหรือคำบรรยายให้อยู่ในรูปของกราฟิกที่ดูแล้วเข้าใจได้ง่าย เช่นกราฟชนิดต่าง ๆ หรือการสร้างแผนผังการจัดองค์กร โปรแกรมประเภทนี้ส่วนมากใช้ในงานธุรกิจซึ่งบางทีก็เรียกว่า Business graphic (ดูข้อเขียนของอาจารย์ชัยยงค์ ในไมโครฯ ฉบับที่ 43) อาจจะใช้อุปกรณ์สำหรับนำภาพออกได้หลายชนิด เช่น ลงแผ่นฟิล์ม, เครื่องลงจุด, เครื่องพิมพ์เลเซอร์ หรือเครื่องพิมพ์สี เป็นต้น ตัวอย่างของโปรแกรมสำเร็จรูป เช่น FreeLance Plus, Graph Plus, Harvard Graphics เป็นต้น

การนำเสนอบนโต๊ะหรือ Desk-Top Presentation ก็เป็นอีกแนวทางหนึ่งของการโปรแกรมประเภทนี้ ซึ่งรวมเอาการนำเสนอทั้งหมดเข้าไว้ด้วยกันทั้งการนำภาพมาเสนอทีละภาพในลักษณะของ Slide Show ซึ่งขีดความสามารถเหล่านี้ส่วนมากจะมีรวมอยู่ในโปรแกรมกราฟิกเพื่อการนำเสนอเกือบทุกตัว

4. โปรแกรมสำหรับการทำภาพจำลองและเคลื่อนไหว (Animation Software)

โปรแกรมประเภทนี้จะเชื่อมโยงและเรียงลำดับภาพเพื่อการสร้างวิดิทัศน์ มีการเปลี่ยนแปลงภาพหนึ่งไปอีกภาพหนึ่งโดยวิธีการต่าง ๆ กัน เช่น ค่อย ๆ เลือนไปหรือลบหายไปทั้งภาพ ส่วนมากรวมเอาภาษาสำหรับการเขียนบท (script) ของภาพเอาไว้ด้วย ซึ่งคุณสามารถที่จะสร้างการเคลื่อนไหวต่าง ๆ และรวมถึงเครื่องมือสำหรับการสร้างภาพกราฟิกอย่างง่าย ๆ การดึงเอาภาพจากโปรแกรมอื่นมาใช้งานก็เป็นขีดความสามารถที่รวมในโปรแกรมประเภทนี้

เนื่องจากขีดความสามารถของซีพียูเอง การทำภาพ Animation โดย IBM PC จึงยังไม่อาจจะนับได้ว่าเป็น Animation อย่างแท้จริง

ส่วนมากจะนิยมใช้สำหรับการสาธิตหรือใช้เป็นติวเตอร์การใช้งานของโปรแกรมต่าง ๆ ตัวอย่างของโปรแกรม เช่น PC Storyboard Plus, Show-Partner F/X, VCN concorde

5. โปรแกรมสำหรับช่วยงานออกแบบ (CADD Program)

โปรแกรมประเภทนี้เป็นลักษณะของโปรแกรมช่วยในการออกแบบ (Computer Aided Design and Drafting) สำหรับงานทางด้านสถาปัตยกรรม วิศวกรรม อุตสาหกรรม และการจัดการทางด้านฟาซิลิตี้ต่าง ๆ ขีดความสามารถของโปรแกรมประเภทนี้จัดอยู่ในชั้นหัวแถว เช่น การใส่มิติ การสร้างภาพ 3 มิติ การแรเงา การทำโมเดลของวัตถุ เป็นต้น ตัวอย่าง

ของโปรแกรม เช่น AutoCAD, Cadkey, Generic CAD

6. โปรแกรมสำหรับงานพิมพ์ (DeskTop Publishing)

DTP คือชื่อของเขาล่ะ และกำลังมาแรงสำหรับการใช้ไมโครคอมพิวเตอร์บนโต๊ะให้เกิดประโยชน์ทางด้านงานพิมพ์ โปรแกรมประเภทนี้นำเอาภาพจากโปรแกรมอื่นที่สร้างเอาไว้ มาประกอบกับข้อความหรือบทความ เพื่อสร้างการพิมพ์โดยเฉพาะ ความสามารถอื่น ๆ ก็เช่นการเปลี่ยนแปลงรูปแบบของอักขระที่จะจัดพิมพ์ การจัดอาร์ตเวอร์กของหน้ากระดาษที่จะพิมพ์ เป็นต้น ตัวอย่างโปรแกรม เช่น IBM Interleaf Publisher, Page Maker, Ventura Publisher

7. เครื่องมือสำหรับการทำกราฟิกอื่น ๆ

โปรแกรมช่วยจัดการทางด้านกราฟิกอื่น ๆ ช่วยทำให้การทำกราฟิกง่ายเข้า เช่น การดึงเอาภาพที่ปรากฏบนจอมาเก็บไว้ในแฟ้มเพื่อนำไปแก้ไข หรือการถ่ายโอนแฟ้มของภาพระหว่างโปรแกรมต่าง ๆ ห้องสมุดสำหรับเก็บภาพต่าง ๆ เพื่อดึงมาใช้ได้ทันที เป็นต้น

การใช้งานในส่วนการทำภาพเคลื่อนไหวที่เรียกว่า Animation นั้น IBM PC ยังไม่มีขีดความสามารถเพียงพอทั้งความสามารถทางด้านกราฟิกฮาร์ดแวร์และ Architecture ของตัวซีพียูเอง เราอาจจะเห็นการ์ตูนบนจอทีวีบ้านเราที่ใช้คอมพิวเตอร์มาช่วยในการเขียน แต่บน Amiga ของ Commadore ไม่ใช่ IBM XT หรือ AT คงทำได้ดีเฉพาะการนำมาทำภาพศิลปะและการทำภาพ slide สวย ๆ เพื่อการนำเสนอเท่านั้น

สำหรับงานด้านการพิมพ์เข้ามามีบทบาทในการใช้ไมโครคอมพิวเตอร์เสมือนหนึ่งโรงพิมพ์บนโต๊ะทำงานของท่าน และกำลังได้รับความสนใจอย่างมาก ท่านอาจจะเคยได้ยินคำว่า เดสก์ท็อปพับลิชชิง เมื่อมีโปรแกรมอย่าง Ventura, Page Maker หรือ GEMS โปรแกรมพวกนี้จัดได้ว่าอยู่ในข่ายที่เรียกว่า WYSIWYG (What you see is what you get) หรือคุณเห็นอย่างไรคุณก็ได้อย่างนั้น นั่นคือเห็นบนจออย่างไรก็ได้อย่างนั้นพิมพ์ออกมา ไม่ว่าจะเป็นการนำเอารูปภาพมาประกอบในหน้าพิมพ์ การเปลี่ยนแปลงรูปแบบของตัวอักษรที่จะพิมพ์ โปรแกรมภาษาไทยหลายบริษัทก็พยายามพัฒนาให้ใกล้ WYSIWYG มากขึ้น นับว่าเป็นความพยายามที่ดี แม้ว่าจะยังห่างไกลมากนัก เพราะภาษาไทยก็มีเรื่องยุ่ง ๆ ที่จะต้องนำมาคิดให้ปวดหัวพอควรอยู่แล้ว

โปรแกรมจัดระบบงานหลายตัวที่ทำงานบน IBM พยายามทำตัวเองให้ใกล้กับ WYSIWYG มากที่สุด นับแต่ WINDOWS ของไมโครซอฟต์ ซึ่งออกมาตั้งแต่พฤศจิกายน ปี 85 อย่างไรก็ตามก็ยังไม่เป็นที่นิยมเท่าที่ควรอันเนื่องมาจากคุณภาพและการใช้หน่วยความจำมากเกินไป DOS4 และ OS/2 ก็พยายามปรับตัวเองเข้าไปในแนวนี้ ซึ่งผมจะนำมากล่าวอีกที ในหัวข้อ GUI หรือ Graphical User Interface ว่าปัญหาและแนวโน้มกำลังไปในทางไหนกัน

ต้องไม่ลืมว่าสิ่งที่ปรากฏบนจอจะปรากฏบนกระดาษหรือรูปแบบอื่นจะสมจริงสมจังได้ จะต้องมีอุปกรณ์พิมพ์ที่มีความสามารถพอกันหรือมากกว่าจึงได้ WYSIWYG อย่างแท้จริง

เทคนิคการแสดงกราฟิกบนจอแบบ Raster Scan

เพื่อให้ทำความเข้าใจเรื่องกราฟิกที่แสดงบนจอให้ดียิ่งขึ้น ผมขอกล่าวอย่างย่อ ๆ ถึงวิธีการนำแสดงผลกราฟิกบนหน้าจอพอเป็นสังเขป แม้ว่ากองบรรณาธิการเคยนำเสนอไปแล้วในไมโครฯ ฉบับที่ 43

เทคนิคการแสดงกราฟิก อาจจะแบ่งได้เป็น 2 วิธีใหญ่ ๆ คือ

1) Bit Mapped
2) Vector

วิธีการของบิตแม็พก็คือ ทุกจุดบนหน้าจอ ซึ่งเรียกกันว่าพิกเซลนั้นมีความสัมพันธ์โดยตรงกับหน่วยความจำแบบหนึ่งต่อหนึ่ง แต่ก็มิได้หมายความว่า 1 พิกเซลจะเป็น 1 บิตในหน่วยความจำเสมอไป อาจจะเป็น 4 บิตหรือ 8 บิต แล้วแต่ว่าจะให้ 1 พิกเซลสามารถแสดงสีหรือระดับความเข้มขาวดำมากน้อยแค่ไหน ข้อสำคัญก็คือจากตำแหน่งของพิกเซลบนหน้าจอเราสามารถมองหาค่าของพิกเซลในหน่วยความจำได้โดยตรง หรือการเปลี่ยนแปลงหน่วยความจำมีผลโดยตรงต่อการแสดงผลบนหน้าจอ เป็นที่เรียกกันติดปากว่าหน่วยความจำที่ใช้ในการแสดงผลคือ วิดีโอแรม

ดังนั้นเทคนิคการแสดงกราฟิกโดยวิธีบิตแม็ต ก็คือการแก้ไขเปลี่ยนหน่วยความจำของการแสดงผลโดยตรง ข้อมูลภาพกราฟิกก็คือข้อมูลที่นำไปเขียนบนหน่วยความจำแสดงผล วิธีการแสดงกราฟิกแบบเวกเตอร์นั้นแตกต่างไปจากวิธีการของบิตแม็ต คือถ้าข้อมูลของภาพไม่ได้อยู่ในรูปที่จะนำไปแสดงบนจอโดยตรง แต่จะอยู่ในรูปของคำสั่งของการลากเส้น เขียนวงกลม เขียนรูปหลายเหลี่ยม และการระบายสีแทน ภาพที่ได้จากการสร้างโดยวิธีของเวกเตอร์จึงเรียกว่า Object-based image คือบ่งบอกลักษณะของภาพมากกว่าที่จะเป็นตัวภาพเอง

จอภาพของการแสดงผลที่ใช้ในไมโครคอมพิวเตอร์ในปัจจุบันจะเป็น Raster Scan คือแสดงในลักษณะกวาดแสดงไปบนจอติดต่อกันจากซ้ายไปขวา โดยใช้หลักการเดียวกันกับทีวี จำนวนเส้นที่กวาดแสดงผลนั้นเป็นตัวกำหนดขีดความสามารถของวิดีโอการ์ด ซึ่งจะ

มีตั้งแต่ 200, 350, 480 จนถึง 800 เส้น แต่เดิมนั้นจอภาพที่แสดงสี CGA สามารถรับสัญญาณภาพให้แสดงผลได้ 200 เส้นใน 1 ภาพหรือที่เรียกว่า Frame Monochrome หรือ Hercules Graphic Adapter ก็ 350 เส้น จะเห็นว่าจำนวนเส้นของการแสดงจะใกล้เคียงกับระบบทีวีที่มีอยู่มาก คือระบบทีวีที่อยู่ในบ้านเราใช้ 625 เส้นใน 2 frame (สำหรับจอแสดงแบบเวกเตอร์นั้นลองย้อนไปดูรายละเอียดในไมโครฯ ฉบับที่ 43)

ผมจะไม่พูดละว่า ภาพบนจอเกิดขึ้นได้อย่างไร เดี๋ยวจะออกนอกแนวทางไปเยอะ เอาเป็นว่า ทุกจุดที่ปรากฏบนจอจะต้องมีหน่วยความจำเก็บค่าของของพิกเซลเอาไว้ นั่นก็คือ การมีหน่วยความจำสำหรับเก็บภาพแบบบิตแม็ตนั่นเอง กราฟิกโดยวิธีการบิตแม็ตจึงสามารถนำแสดงบนหน้าจอได้ทันที ส่วนกราฟิกแบบเวกเตอร์นั้นจะต้องให้ซีพียูแปลงออกมาเป็นบิตแม็ตในหน่วยความจำของการแสดงผลเสียก่อน จึงจะเห็นภาพปรากฏบนหน้าจอได้

ถ้าเปรียบเทียบให้เห็นชัดถึงความแตกต่างระหว่างบิตแม็ตและเวกเตอร์ก็คงจะพอประมาณได้กับงานศิลปะ 2 วิธีคือ การวาดภาพ จิตรกรรม และงานปฏิมากรรม อย่างแรกนั้นเป็นการสร้างภาพขึ้นมาเปรียบได้กับกราฟิกชนิดบิตแม็ต อย่างที่สองเป็นการสร้างงานศิลปะขึ้นมา สามารถที่จะมองในแง่มุมที่แตกต่างกันได้ ซึ่งเปรียบได้กับกราฟิกแบบเวกเตอร์ อย่างที่สองเป็นการสร้างงานศิลปะขึ้นมา สามารถที่จะมองในแง่มุมที่แตกต่างกันได้ ซึ่งเปรียบได้กับกราฟิกแบบเวกเตอร์

งานกราฟิกที่ใช้ทางด้านสถาปนิกและวิศวกรรมจะต้องเป็นชนิดเวกเตอร์ คือจะต้องมีทั้ง 3 มิติและสามารถใช้ขีดความสามารถของซีพียูแปลงเวกเตอร์ของกราฟิกให้ปรากฏออกมาบนจอได้ในแง่มุมที่แตกต่างกัน โปรแกรม AutoCAD เป็นตัวอย่างหนึ่งของการสร้างภาพแบบเวกเตอร์ ซึ่งสามารถกำหนดแง่มุมของการมองได้ ไม่เพียงแต่เท่านั้นกราฟิกในรูปแบบของเวกเตอร์สามารถที่จะนำมาแสดงในลักษณะย่อหรือขยายในอัตราส่วนเท่าไหร่ก็ได้ โดยที่ภาพที่แสดงออกยังคงรูปร่างที่เหมือนกับความเป็นจริง

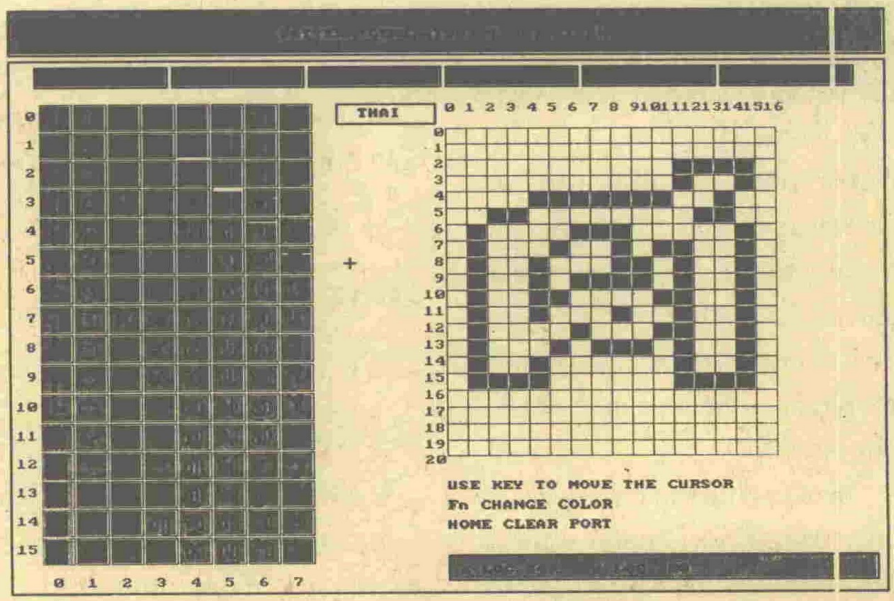
เพื่อให้ผู้อ่านได้เห็นภาพได้ชัดเจนระหว่างบิตแม็ตและเวกเตอร์ลองมาดูการแสดงตัวหนังสือบนจอกราฟิก โดยมากการแสดงอักขระบนหน้าจอจะเป็นแบบบิตแม็ต อักษรแต่ละตัวถูกกำหนดให้มีหน้าตาต่างกัน และเข้าไปครอบครองเนื้อที่ในหน่วยความจำเท่า ๆ กัน การนำอักขระเหล่านี้มาแสดงจึงไม่น่าจะผิดที่จะเรียกว่า การพิมพ์ตัวอักษรบนจอ ถ้าเราจะขยายการพิมพ์ให้ใหญ่ขึ้นก็จะต้องขยายให้เป็นอัตราส่วนที่เป็นเลขจำนวนเต็มเท่านั้น ภาพของอักษรที่ปรากฏบนหน้าจอไม่ได้มีความละเอียดเพิ่มขึ้น แต่กลับจะลดถอยลงไปมองเห็นรอยยัก ๆ บนตัวอักษรได้ชัดเจนยิ่งขึ้น

ในทางตรงกันข้ามการแสดงอักขระบนจอแบบเวกเตอร์นั้นเป็นการบอกวิธีการเขียนตัวอักษรบนจอ เช่น จากจุดเริ่มต้นลากเส้นขึ้นข้างบน 4 หน่วย ลากไปทางขวา 3 หน่วย ลากลง 4 หน่วย ดังนี้เป็นต้น ลักษณะเช่นนี้จึงไม่ผิดที่จะเรียกว่าเป็นการสอนคอมพิวเตอร์ให้รู้จักเขียนหนังสือ

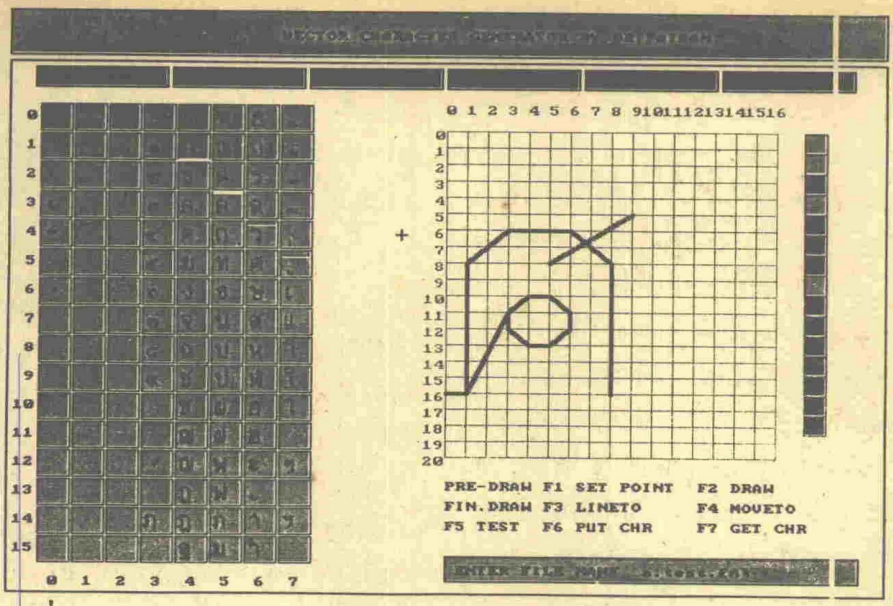
ข้อดีของการเขียนแบบเวกเตอร์ก็คือ สามารถขยายโดยอัตราส่วนเท่าใดก็ได้ โดยไม่เสียความละเอียดของแง่มุม สามารถจะเขียนให้ทิศทางใด ๆ ก็ได้ไม่จำเป็นต้องเขียนจากซ้ายไปขวา อาจจะเขียนให้เอียงเป็นมุมเท่าใดก็ได้

ผมมีตัวอย่างโปรแกรมการสร้างรูปแบบของตัวอักษรมาให้เปรียบเทียบกัน 2 รูป คือ รูปที่ 1 การสร้างอักขระในลักษณะของบิตแม็ต จะเห็นว่าเมื่อขยายออกมาจะมองรอยต่อจุดต่าง ๆ ที่ไม่เรียบได้ชัดเจน ในรูปที่ 2 เป็นการสร้างรูปแบบของอักขระในแบบของเวกเตอร์ เมื่อขยายออกมาทางขวามือ ก็ยังคงรักษารายละเอียดของรูปแบบได้อย่างครบถ้วน

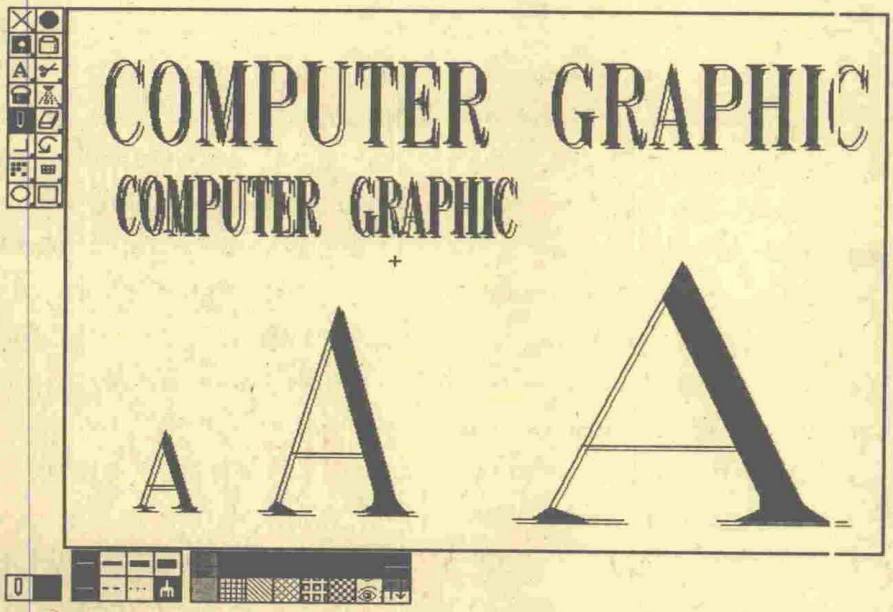
การใช้งานของวิธีการสร้างภาพกราฟิกบนจออาจจะนำมาใช้ผสมกันทั้งสองแบบคือ บิตแม็พและเวกเตอร์ แต่เวลาเก็บผลงานคือภาพที่ได้ออกมาเอาไปเก็บแบบบิตแม็พ วิธีการเช่นนี้



รูปที่ 1 การกำหนดรูปแบบของอักขระ (Font) โดยวิธีบิตแม็ต



รูปที่ 2 การกำหนดรูปแบบของอักขระในลักษณะของเวกเตอร์



รูปที่ 3 การเขียนอักขระแบบเวกเตอร์ของโปรแกรม Dr.halo จะเห็นความละเอียดของเหลี่ยมมุมยังคงเดิม ไม่ว่าจะขยายขนาดเท่าไหร่ก็ตาม

ใช้ในโปรแกรมสำเร็จรูปในการสร้างกราฟิกเพื่อการนำเสนอ อย่างเช่น PC Paint, PC PaintBrush หรือ Dr.-halo เป็นต้น

อุปกรณ์ที่เกี่ยวข้องกับคอมพิวเตอร์กราฟิก

ก่อนผมจะพูดถึงการนำคอมพิวเตอร์กราฟิกไปใช้งานในแต่ละประเภท ขอพูดถึงอุปกรณ์ที่เกี่ยวข้องไว้เสียตรงนี้เลย

ถ้าหากจะแบ่งอุปกรณ์ที่เกี่ยวข้องกับคอมพิวเตอร์กราฟิกก็เห็นจะแบ่งประเภท ดังนี้

1. ตัวไมโครคอมพิวเตอร์เอง ซึ่งบางทีก็ใช้คำว่า Graphic Engine (คือฝรั่งก็ยังติดคำว่าเครื่องยนต์อยู่ อะไรที่ทำงานได้ก็จะถือว่าเป็นเครื่องยนต์หมด นี่ถ้าเอดิสันเกิดก่อนเจมส์วัตต์ คงไม่ถือคำว่า Engine เป็นสรณะ)

2. อุปกรณ์แสดงภาพ อันได้แก่ การ์ดแสดงผลและตัวจอด้วย

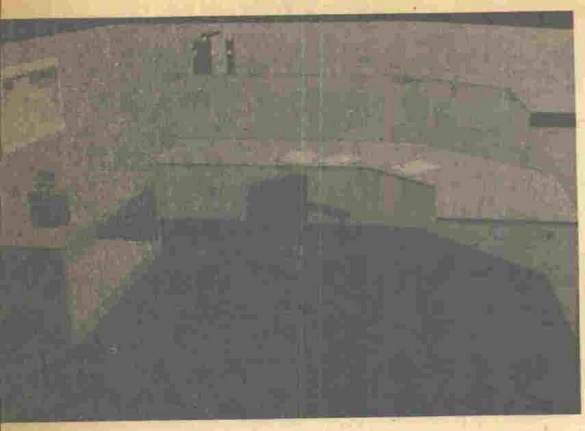
3. อุปกรณ์พิมพ์รูปกราฟิก

4. อุปกรณ์นำภาพจากภายนอกเข้าไปยังคอมพิวเตอร์

5. อุปกรณ์ช่วยในการเปลี่ยน-แปลงแก้ไขภาพ

เครื่องยนต์สร้างกราฟิก (Graphic Engine)

ในส่วนของตัวไมโครคอมพิวเตอร์นั้น ซีพียูมีส่วนสำคัญอย่างมากในการสร้างภาพ โดยเฉพาะการสร้างภาพที่ใช้เทคนิคแบบเวกเตอร์ เพราะซีพียูเองต้องทำการคำนวณเพื่อแปลงเวกเตอร์ออกมาเป็นภาพในลักษณะบิต-แม็พ เพื่อให้จอภาพแสดงผลได้ สถาปัตยกรรมของตัวโปรเซสเซอร์ใช้ใน IBM PC ไม่ว่าจะเป็น 80286, 80386 หรือ 80486 ที่กำลังจะมา ไม่ได้ออกแบบสำหรับการทำงานเพื่อการทำกราฟิกโดยเฉพาะ ดังนั้นถ้าจะให้สามารถสร้างภาพกราฟิกได้เร็วยิ่งขึ้น จึงจำเป็นต้องมีโคโปรเซสเซอร์มาช่วยลดงานของซีพียู การ์ดแสดงผลจะต้องมีความสามารถมากกว่าในปัจจุบันที่มีอยู่ แม้ว่าการ์ด VGA หรือ Super VGA จะแสดงภาพได้ละเอียดก็ช่วยซีพียูในส่วนของการแสดงบิตแม็พเท่านั้น ยกตัวอย่างให้เห็นง่าย ๆ ในโปรแกรม AutoCAD สมมติว่าเราสั่งให้หมุนมุมมองวัตถุไป 30 องศา ซีพียูต้องทำหน้าที่คำนวณพิกเซลที่จะปรากฏบนหน้าจอใหม่หมดทั้งหน้าจอ เสียเวลาในการคำนวณ (เพื่อแปลงเวกเตอร์ออกมาเป็นบิตแม็พ)มาก ถ้าตัวการ์ดของการแสดงผลมีความสามารถมากกว่านี้ก็จะทำให้ช่วยลดเวลาการทำงานของซีพียู (และเพิ่มความเร็วในการทำงาน)ได้มาก ไมโครโปรเซสเซอร์สำหรับการแสดงรุ่นใหม่หลายตัว เช่น TMS 34020 จาก Texas instrument หรือ 82786 จาก Intel มีขีดความสามารถในการแสดงผลใน



ลักษณะฉลาดมากขึ้น เช่น สามารถออกคำสั่งให้เขียนรูป Polygon ได้ สามารถระบายสีในกรอบอันใดอันหนึ่งเป็นต้น เฉพาะสองอย่างนี้ก็ลดเวลาการทำงานของซีพียูลงไปได้มาก อีกสักปีสองปี เราคงจะได้เห็นโคโปรเซสเซอร์อันฉลาด (ยังไม่ถึงกับชาญฉลาด) พวกนี้บนการ์ดแสดงผลมากขึ้น เท่าที่ผมพบในขณะนี้มีการ์ดแสดงผลของ Tseng Lab EVA480 พยายามเพิ่มขีดความสามารถในการขยายภาพบิตแม็พได้ 2 เท่า 4 เท่า และ 8 เท่า ทำให้การขยายภาพบิตแม็พได้รวดเร็วโดยไม่ต้องพึ่งพาซีพียูให้เสียเวลา แต่ก็ยังไม่นับว่าเป็นโคโปรเซสเซอร์เพื่อการแสดงผู้ฉลาด

สำหรับตัวการ์ดแสดงผล ผมจะไม่ขอกล่าวมากเพราะมีบทความเกี่ยวกับเรื่องนี้ในฉบับนี้อยู่แล้ว แต่ก่อนที่ท่านจะตัดสินใจซื้อการ์ด VGA สำหรับเครื่องของท่าน ควรจะจำเอาไว้ว่า แต่ละการ์ดทำงานได้ครบไปกันได้ (compatible) กับ VGA ของ PS/2 แต่ก็ไม่ได้หมายความว่าประสิทธิภาพในการทำงานจะเท่ากัน บางการ์ดอาจจะเร็วกว่าหรือช้ากว่า IBM PS/2 ซึ่งเป็นต้นตำรับ VGA บางการ์ดก็มีความสามารถพิเศษที่นอกเหนือจาก VGA เป็นต้นว่า มีความละเอียดสูงถึง 640 × 800 จุด หรือขยายภาพได้ทันทีทันใด เป็นต้น ความสามารถพิเศษเช่นนี้จะไม่ก่อประโยชน์อันใดแก่ผู้ใช้เลย ถ้าหากว่าไม่มีโปรแกรมสำเร็จรูปตัวไหนออกแบบมาเพื่อใช้มัน นอกเสียจากว่าคุณจะเขียนโปรแกรมควบคุมมันเสียเอง ซึ่งก็ต้องมีคู่มือทางเทคนิคประกอบ ผู้ผลิตการ์ดเหล่านี้ก็คงตระหนักดีในข้อนี้ก็พยายามเขียนโปรแกรมไดรเวอร์สำหรับโปรแกรมสำเร็จรูปต่าง ๆ ให้อาจจะ 3 หรือ 4 โปรแกรม เช่น ไดรเวอร์สำหรับ ACAD สำหรับ Windows หรือสำหรับ WordPerfect version 5 ที่นอกเหนือจากนั้นคุณก็จะได้ขีดความสามารถแค่เพียง VGA ของ PS/2 เท่านั้น เพราะฉะนั้นก่อนพิจารณาซื้อ ควรทำความเข้าใจลักษณะ เฉพาะตัวของแต่ละการ์ดให้ดีเสียก่อน

สำหรับอุปกรณ์การแสดงภาพนั้น มอนิเตอร์จะเป็นอุปกรณ์หลัก ความแตกต่างของจอมอนิเตอร์มีผิดแผกกันไปบ้าง จากของเดิมที่เคยมีตั้งแต่การ์ดแสดงผลขาวดำและ CGA เป็น EGA และ VGA มอนิเตอร์ที่ใช้สำหรับการ์ดแต่ละชนิดจะไม่เหมือนกัน ทั้งไม่ถึงเป็นเช่นนั้น ถ้าจะบอกว่าก็ความละเอียดของการแสดงผลของแต่ละการ์ดไม่เท่ากันก็จะเป็นการตอบแบบกำปั้นทุบก้อนกรวด (ที่ยังไม่ทันเป็นดิน) เกินไป ลองมานึกดูว่าความละเอียดของภาพที่สร้างออกมาในลักษณะของสัญญาณทางไฟฟ้าจากการ์ดแสดงผลส่งให้มอนิเตอร์ มันแตกต่างกันยังไง เมื่อความละเอียดของภาพสูงขึ้น เริ่มตั้งแต่สายตาคนเราจะไม่สามารถมองเห็นการเปลี่ยนแปลงของภาพที่เร็วกว่า 15 ภาพต่อวินาที ภาพที่เราเห็นบนทีวีทุกวันมีการเปลี่ยนแปลงของภาพประมาณ 30 ภาพต่อวินาที ลองเปรียบเทียบบรรญัติไตรยางค์เล่น ๆ ดู ถ้าหาก 1 ภาพ มีการกวาดแสดงผลจากซ้ายไปขวา 625 เส้น ดังนั้นการกวาดแสดงผล 1 เส้นบนจอภาพจะต้องใช้เวลา 1/(30 × 625) วินาที หลักการกวาดแสดงผลดังกล่าวก็นำมาใช้กับมอนิเตอร์ของคอมพิวเตอร์เช่นกัน ยังตอบคำถามไม่ตรงประเด็นเลย งั้นขยายความต่ออีกนิด ในสัญญาณภาพ (Video) ที่ส่งไปยังมอนิเตอร์นั้นมีสัญญาณซึ่งกำหนดจุดเริ่มต้นของภาพ ซึ่งเรียกว่า Vertical Sync หรือ V.Sync และจุดเริ่มต้นของการกวาดแสดงผลในแต่ละเส้นภายในภาพ ซึ่งเรียกว่า Horizontal Sync หรือ H.Sync

ความถี่ของ V.Sync เป็นตัวกำหนดจำนวนภาพต่อวินาทีซึ่งจะต่ำกว่า 15 ภาพต่อวินาทีไม่ได้ เพราะจะทำให้สายตามองเห็นการกะพริบบนหน้าจอ ซึ่งส่วนมากจะให้ค่าประมาณ 30 ภาพต่อวินาที ความถี่ของ H.Sync ก็เป็นอัตราส่วนกับจำนวนที่แสดงภายใน 1 ภาพ คราวนี้คงจะพอเห็นแล้วว่าความละเอียดของภาพต่างกัน ความถี่ของ H.Sync ก็จะต่างกัน จอรุ่นก่อนสร้างภาพด้วย H.Sync ค่าหนึ่งซึ่งเหมาะสำหรับ CGA เท่านั้น เมื่อการ์ดแบบ EGA ซึ่งมีความละเอียด 350 เส้น ความละเอียดเพิ่มขึ้นอีกเกือบเท่าตัว เมื่อเทียบกับ 200 เส้นใน CGA H. Sync ก็ต้องเพิ่มอีกเกือบเท่าตัว จอเดิมซึ่งทำงานกับ CGA มานานรับไม่ไหวแน่ ก็ต้องเป็นจอที่สามารถรับสัญญาณ EGA ได้เท่านั้น เอาอีกละสิ มีการ์ด VGA ออกมาอีก คราวนี้ความละเอียดในแนวดิ่งถึง 480 เส้น ความถี่ของ H.Sync สูงขึ้นไปอีก จอ EGA ก็รับไม่ไหวก็ต้องเป็นจอ VGA เท่านั้น เห็นไหมละครับ การ์ดแต่ละชนิดก็ต้องการมอนิเตอร์ที่แตกต่างกัน ปัญหายังไม่จบเพียงแค่นั้น เพราะ VGA สามารถทำตัวเองเสมือนหนึ่งแบบ EGA หรือ CGA ก็ได้ เพื่อที่โปรแกรมรุ่นเก่า ๆ จะสามารถทำงานได้ แล้วยังงี้เรามิต้องคอยเปลี่ยนมอนิเตอร์บ่อย ๆ หรือ เวลาเราเปลี่ยนโปรแกรมที่มีโหมดการแสดงผลที่ไม่เหมือนกัน คำตอบก็คือ แน่นอนคุณจะต้องเปลี่ยนถ้าหากมอนิเตอร์ของคุณรับได้เฉพาะอย่าง ผมกำลังพาคุณผู้อ่านไปพบกับมอนิเตอร์ระบบ Multisync หรือ Sync ได้หลายรูปแบบ ถ้าคุณมีมอนิเตอร์ชนิดนี้คุณก็ไม่ต้อง

เปลี่ยน Monitor ทุกครั้งที่จะใช้โปรแกรมเก่าที่ยังใช้ CGA หรือ EGA อยู่

ขอต่อเรื่องมอนิเตอร์อีกนิดหนึ่ง ตรงที่ว่าการ์ด VGA สามารถกำหนดให้พิกเซลหนึ่งมีสีได้มากกว่า 4096 สี และแสดงสีบนหน้าจอเดียวกันได้สูงถึง 256 สี จุดนี้มันจะไม่เกี่ยวข้องกับเรื่องจอภาพนิดหนึ่ง เกี่ยวข้องยังไงลองอ่านต่อดูซิครับ ปกติสัญญาณที่ป้อนเข้ามอนิเตอร์ที่จะกำหนดสีได้จะมี 4 สัญญาณที่สำคัญ คือ สัญญาณสีแดง สีเขียว สีน้ำเงิน และสัญญาณความเข้ม ในความเป็นจริงแล้วสัญญาณดิจิตอล 4 เส้นจะสามารถกำหนดความแตกต่างได้แค่ 16 ชนิดเท่านั้น แล้วจะแสดง 256 สีได้อย่างไร คำตอบก็คือ VGA ไม่ได้ส่งสัญญาณมาแบบดิจิตอลแต่สั่งมาแบบอะนาลอก นั่นคือระดับของสัญญาณทั้ง 4 ไม่ได้มีแค่ 2 ระดับ แต่มีได้อนันต์ระดับตั้งแต่ 0 Volt ถึง 5 Volt (รายละเอียดขอให้อ่านในเรื่องของ VGA) ดังนั้นมอนิเตอร์นิเตอร์ก็จะต้องสามารถรับสัญญาณอะนาลอกได้ด้วย ปกติมอนิเตอร์ CGA หรือ EGA จะใช้ลอจิกเกตเป็นตัวรับสัญญาณจากการ์ดแสดงผล สัญญาณอะนาลอกป้อนเข้าก็แปลงเป็นดิจิตอลหมด โดยเอาเกณฑ์ของการเปลี่ยนแปลงระดับลอจิกของเกตตัวที่ใช้รับสัญญาณเป็นหลัก

ดังนั้นการซื้อมอนิเตอร์สำหรับ VGA นอกจากจะดูว่าเป็น MultiSync แล้วก็ต้องดูว่ารับสัญญาณอะนาลอกได้หรือไม่ด้วย

อุปกรณ์การพิมพ์กราฟิก

ถัดจากเรื่องของการแสดงผลบนจอก็มาดูเรื่องของการนำภาพจากคอมพิวเตอร์มาพิมพ์

การนำภาพกราฟิกออกอาจจะใช้อุปกรณ์ได้หลายชนิด เช่น

- เครื่องพิมพ์ ทั้งดอตแมตริกซ์หรือเลเซอร์พรินเตอร์
- เครื่องลงจุดหรือพล็อตเตอร์
- เครื่องเปลี่ยนภาพเป็นฟิล์ม Silde 35 mm.

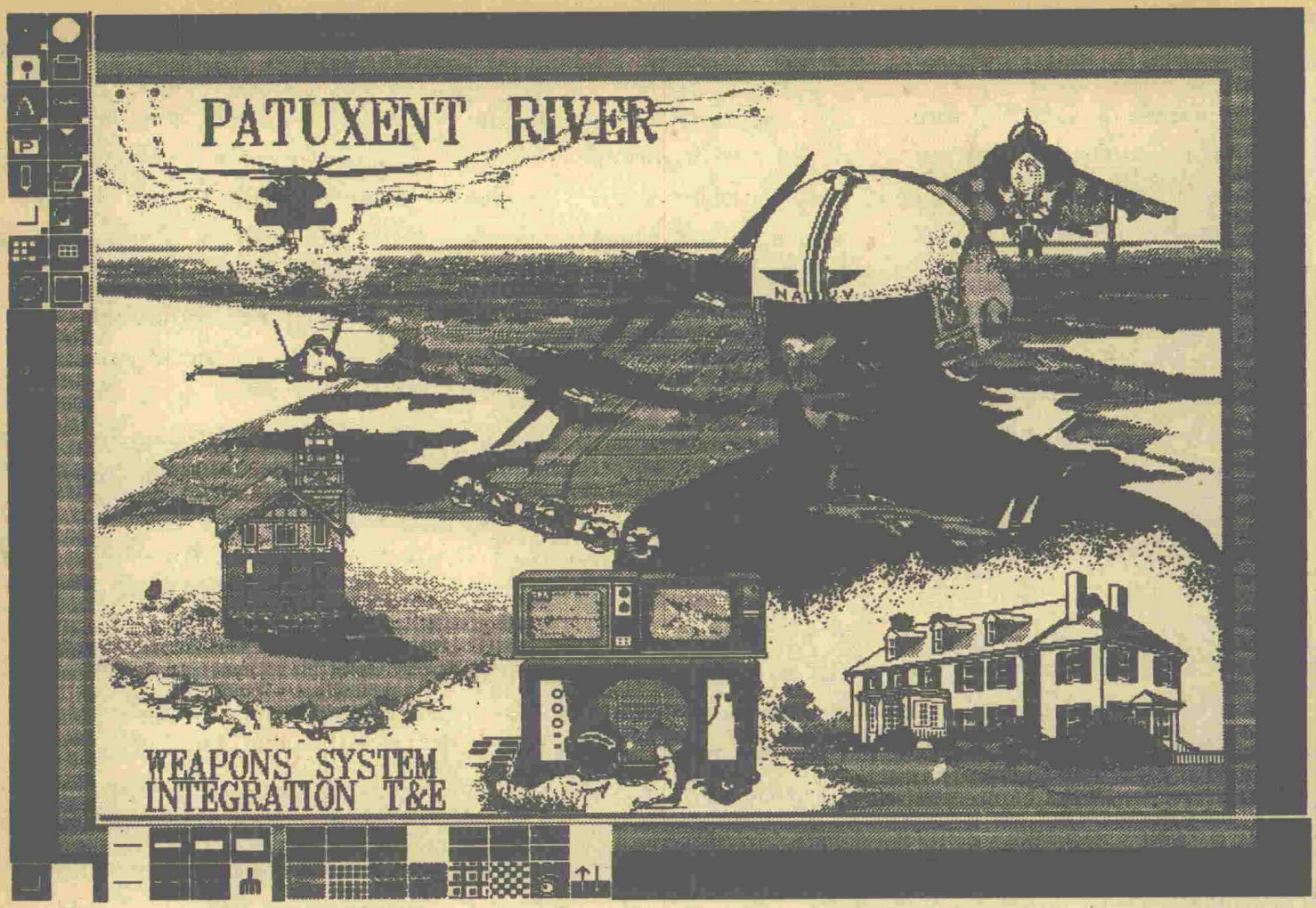
พูดถึงเครื่องพิมพ์ แน่นอนต้องเป็นแบบดอตแมตริกซ์ จึงจะสามารถพิมพ์กราฟิกได้ เครื่องพิมพ์สมัยนี้พัฒนาไปไกลและราคาถูกลงมาก ความละเอียดของภาพก็สามารถมีได้ตั้งแต่ 60 จุดต่อนิ้วไปจนถึง 480 จุดต่อนิ้ว ต้องทำความเข้าใจตรงนี้ให้ดีนะครับ ความละเอียดที่ว่านี้เป็นความละเอียดในแนวราบเท่านั้นนะครับ แล้วก็ไม่ได้หมายความว่าจะกำหนดให้เป็นเท่าไหร่ก็ได้ระหว่าง 60 จุดต่อนิ้วจนถึงสูงสุด 480 จุดต่อนิ้ว แต่จะเป็นชั้น ๆ อย่างมาก 7 ชั้น เช่น เครื่องพิมพ์ราคาถูกอย่าง Super 5 หรือ LX series ของ Epson กำหนดความละเอียดในแนวราบได้ 60, 72, 80, 90, 120 และ 240 จุดต่อนิ้ว สำหรับในแนวดิ่งนั้นค่อนข้างจะตายตัวเนื่องจากหัวพิมพ์จะมีการเรียงจุดพิมพ์หรือพินไว้ตายตัวสำหรับเครื่องพิมพ์ชนิด 9 จุด คือประเภทราคาถูกทั่ว ๆ ไป แต่ละจุดพิมพ์จะห่างกัน 1/96 นิ้ว สำหรับเครื่องพิมพ์ชนิด 24 พิน × แต่ละจุดจะวางเรียงห่างกันในแนวดิ่ง 1/(3 × 96 นิ้ว) การพิมพ์กราฟิกให้ได้สัดส่วนจะต้องมีการปรับความละเอียดในแนวราบและแนวดิ่งให้ใกล้เคียงกันจึงจะได้ภาพที่สมสัดส่วน เครื่องพิมพ์แบบดอตแมตริกซ์จึงสามารถกำหนดรายละเอียดได้ค่อนข้างตายตัว

สำหรับเครื่องพิมพ์ที่กำลังมีบทบาทย่างสูงสำหรับกราฟิก ก็คือ เครื่องพิมพ์ใช้ลำแสงเลเซอร์ การกำหนดความละเอียดของภาพก็เข้าทำนองเดียวกับเครื่องพิมพ์ชนิดดอตแมตริกซ์ แต่ว่าทั้งแนวดิ่งและแนวราบจะมีความละเอียดเหมือนกันอย่างเช่น Laser Jet + ของ HP สามารถกำหนดความละเอียดได้ 3 ขั้น คือ 75, 150 และ 300 จุดต่อนิ้วเท่านั้น ความเร็วในการพิมพ์โดย Laser Jet + ไม่ได้เร็วไปกว่าเครื่องดอตแมตริกซ์เลย ตัวอย่างเช่น รูปที่ 4 เป็นการพิมพ์รูปจากที่เราคุ้นหน้าคุ้นตาโดยเครื่องพิมพ์ NEC P2200 ที่ความละเอียด 240 จุดต่อนิ้ว ใช้เวลาในการพิมพ์ประมาณ 2 นาที รูปที่ 5 เป็นการพิมพ์ภาพเดียวกันโดย Laser Jet + ความละเอียดในการพิมพ์ 300 จุดต่อนิ้ว ใช้เวลาในการพิมพ์ภาพแรกประมาณ 3.5 นาที ข้อได้เปรียบของ Laser jer + อยู่ตรงไหนล่ะ ? สังเกตุดูคุณภาพให้ดี หมึกลงสม่ำเสมอดีกว่า เครื่องพิมพ์แบบดอตแมตริกซ์ ซึ่งจะมีปัญหามากถ้าผ้าหมึกถูกใช้ไปนาน ๆ

โชคร้ายสำหรับผู้ใช้เครื่องพิมพ์เสียจริง ๆ คำสั่งควบคุมเครื่องพิมพ์ของแต่ละบริษัทไม่เหมือนกันเสียทีเดียว แม้ว่าจะใกล้เคียงกันอยู่บ้างแต่ถ้าเป็นการพิมพ์กราฟิกแล้วผิดไม่ได้เลย พ่อเจ้าประคุณเอย! พิมพ์ขยะมาให้กองโตเชียวละ การเลือกเครื่องพิมพ์สำหรับโปรแกรมสำเร็จรูปทางด้านกราฟิกจึงมีความสำคัญอย่างยิ่ง แม้ว่าผู้ทำซอฟต์แวร์จะพยายามเอาใจผู้ซื้อโดยมีไดรเวอร์สำหรับพรินเตอร์ให้เลือกมากมาย แต่ก็ไม่ครบไปเสียทุกยี่ห้อหรอก โดยเฉพาะโปรแกรมที่เขาพัฒนาที่อเมริกา ผู้ใช้เขาใช้เครื่องพิมพ์ที่ส่วนใหญ่ไม่มียี่ห้อที่ตรงกับของเรา หรือเครื่องพิมพ์ที่เพิ่งออกมาใหม่ยังไม่อยู่ในรายการที่มีให้เลือก เราก็ต้องลองผิดลองถูกจนกว่าจะพิมพ์ออกมาได้ อาจจะต้องใช้เวลาหลายชั่วโมง บางครั้งก็อาจจะไม่สำเร็จผล

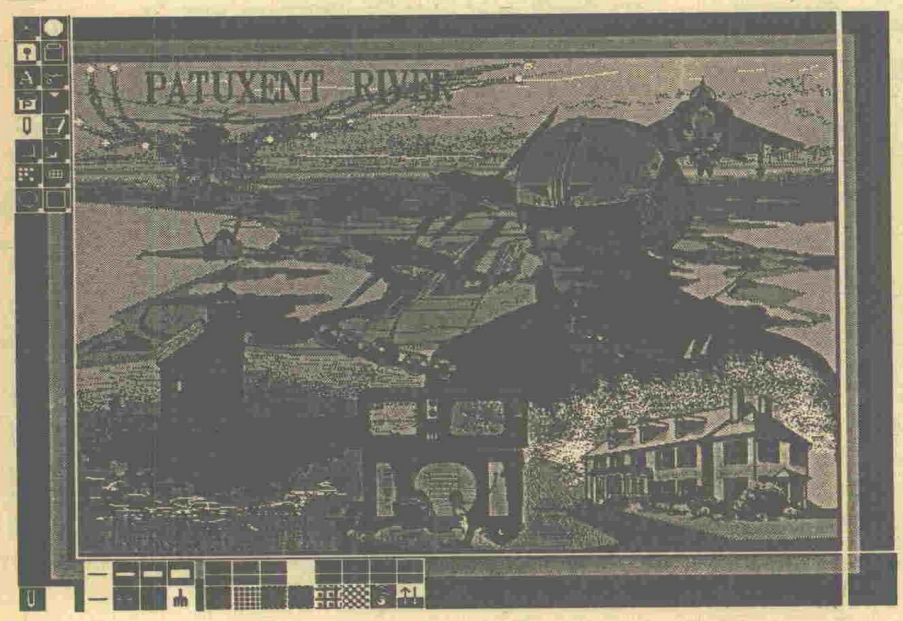
สำหรับเครื่องพิมพ์ที่ใช้เลเซอร์คงจะมีราคาถูกลงอย่างแน่นอน เพราะหลายบริษัทเริ่มทำการแข่งทำออกมาขาย โดยมี HP เป็นตัวยืน ตามด้วย Cannon, NEC และล่าสุด Epson ก็เอากับเขาแล้วออก EPL 6000 มาหวังขอส่วนแบ่งตลาด

ภาพบนจอเป็นสีสวยงามแต่ปรากฏบนกระดาษพิมพ์เป็นขาวดำ ออกจะเสียความรู้สึกเอามาก ๆ สำหรับภาพศิลปะที่อุตส่าห์ทำบนเครื่องหลายชั่วโมง อย่าเพิ่งท้อครับเครื่องพิมพ์สีก็มีครับ ทั้งแบบดอตเมตริกซ์ และเลเซอร์เจ็ต สำหรับ PAINT JET และ HP นั้นเป็นเครื่องพิมพ์ที่สามารถพิมพ์สีได้ถึง 1024 สี



รูปที่ 4 พิมพ์กราฟิกโดย NEC P2200 ความละเอียด 240 จุดต่อนิ้ว

สนนราคากลับถูกกว่า Laser jet + ที่พิมพ์ขาวดำเสียอีก เพราะอะไรหรือครับ



รูปที่ 5 พิมพ์กราฟิกโดย Laser Jet II ความละเอียด 300 จุดต่อนิ้ว

ก็เพราะความเร็วในการพิมพ์ยิ่งแย่ลงไปอีก ลองคิดดูแผ่นหนึ่งใช้เวลาพิมพ์ 8 นาที หรือมากกว่านั้นขึ้นอยู่กับว่าใช้สีมากน้อยแค่ไหน กระดาษที่ใช้พิมพ์ก็ต้องแบบกระดาษพิเศษ ใช้กระดาษธรรมดาไม่ได้ ตกแล้วค่าพิมพ์ต่อแผ่นไม่ต่ำกว่า 20 บาท สำหรับผมคิดว่าคงจะต้องรอให้ประสิทธิภาพการทำงานที่ดีกว่านี้หน่อย จึงจะพิจารณา (แม้ว่าพิจารณาแล้วซื้อไม่ลงก็ตาม)

อีกวิธีหนึ่งของการใช้รูปกราฟิกบนกระดาษก็คือ การใช้เครื่องพล็อตเตอร์ พล็อตเตอร์ตั้งแต่ขนาดเล็กใช้ได้แค่กระดาษ A4 ไปจนถึงขนาดที่ใช้ได้กับกระดาษสำหรับการเขียนแบบโดยเฉพาะ พล็อตเตอร์ซึ่งบางทีก็เรียกว่า x-y plotter นี้ใช้ได้เฉพาะกับรูปกราฟิกในรูปแบบของเวกเตอร์เท่านั้น จะใช้กับบิตแม็พไม่ได้เลย

การบันทึกภาพลงฟิล์ม (Digital film Recorder)

การนำภาพออกจากคอมพิวเตอร์อีกรูปแบบหนึ่งก็คือ การถ่ายภาพลงยังแผ่นฟิล์ม 35 mm. ซึ่งอาจจะเป็นสไลด์หรือเนกาตีฟก็ได้ วิธีการนี้เป็นที่นิยมกันมากสำหรับการทำกราฟิกเพื่อการนำเสนอ (Presentation) มีผู้ประมาณกันว่าในอเมริกาธุรกิจในการนำเสนอในปี 88 อยู่ในวงเงินถึง 770 ล้านดอลลาร์ (จาก PC Magazine ฉบับเดือนกุมภาพันธ์ 89) ซึ่งนับว่าเป็นธุรกิจที่ใหญ่โตไม่ใช่น้อยเลย อุปกรณ์ที่ใช้ในการทำสไลด์มีราคาตั้งแต่ 3,000 เหรียญ จนถึง 15,000 เหรียญ หลายท่านอาจจะสงสัยว่าทำไมราคามันแพงนัก ไม่เห็นจะมีอะไรยุ่งยากเลย หาซื้อกล้องมาสักตัว สวมขาสำหรับตั้งกล้องสักอันก็สามารถถ่ายภาพจากหน้าจอได้แล้ว ผมไม่ได้แย้งคือ ผมไม่เถียง แต่คุณภาพของภาพที่ได้ก็จะมีความละเอียดของภาพไม่ดีไปกว่าคุณภาพของจอที่กำลังแสดงอยู่ VGA ก็ 640 × 480 จุด แต่อุปกรณ์ที่ผมกล่าวถึงมีความละเอียดมากกว่าจอมากนัก

ลักษณะการทำงานของอุปกรณ์ดังกล่าวก็คือ การนำข้อมูลทางกราฟิกมานำเสนอบนจอ CRT ซึ่งอยู่ในกล่องปิดสนิทต่อหน้ากล้องถ่ายภาพ CRT ที่ว่าเป็น CRT ที่มีความละเอียดสูง ซึ่งอาจจะสูงถึง 4,096 จุดในหนึ่งเส้น CRT ที่มีความละเอียดสูงเช่นนี้ก็ต้องเป็นแค่ CRT ขาวดำเท่านั้น แล้วจะสร้างให้เกิดเป็นสีบนสไลด์อย่างไงละครับ เทคนิคก็คือการถ่าย 4 หน โดยการใช้ฟิลเตอร์กรองแสงเข้าช่วยทีละสี คือ แดง เขียว น้ำเงิน และขาว โดยการแยกเอาสัญญาณทีละสีมาแสดงบน CRT แล้วก็ใช้ฟิลเตอร์กรองแสงทีละสี บวกกับความเข้มของสัญญาณภาพเป็น 4 ว่ากันกว่าจะได้ภาพออกมาแต่ละภาพใช้เวลามากกว่าชั่วโมง บางยี่ห้อใช้เวลาข้ามคืน ขณะนี้มีสินค้าประเภทนี้ในท้องตลาด (อเมริกัน) ประมาณ 20 บริษัท

การใช้งานอุปกรณ์ถ่ายกราฟิกลงสไลด์มิใช่ว่าจะนำกราฟิกตัวไหนลงก็ได้เหมือนกับถ่ายภาพจากหน้าจอ แต่ว่าจะต้องมีโปรแกรมสำหรับการควบคุม นำรูปกราฟิกที่อยู่ในแฟ้มไปสร้างใหม่บนจอ CRT ของเครื่องถ่ายลงสไลด์ มีโปรแกรมกราฟิกหลายตัวที่สนับสนุนการทำงานนี้โดยตรง

อุปกรณ์นำภาพเข้า

บางครั้งการนำภาพจากภายนอกเข้าไปยังเครื่องโดยตรง ลดงานการสร้างภาพกราฟิกอย่างมาก ไม่ว่าจะเป็นภาพลายเส้นหรือภาพจริง ๆ เพื่องานศิลปะหรือเพื่อการเก็บบันทึกภาพทำเป็นฐานข้อมูลก็แล้วแต่

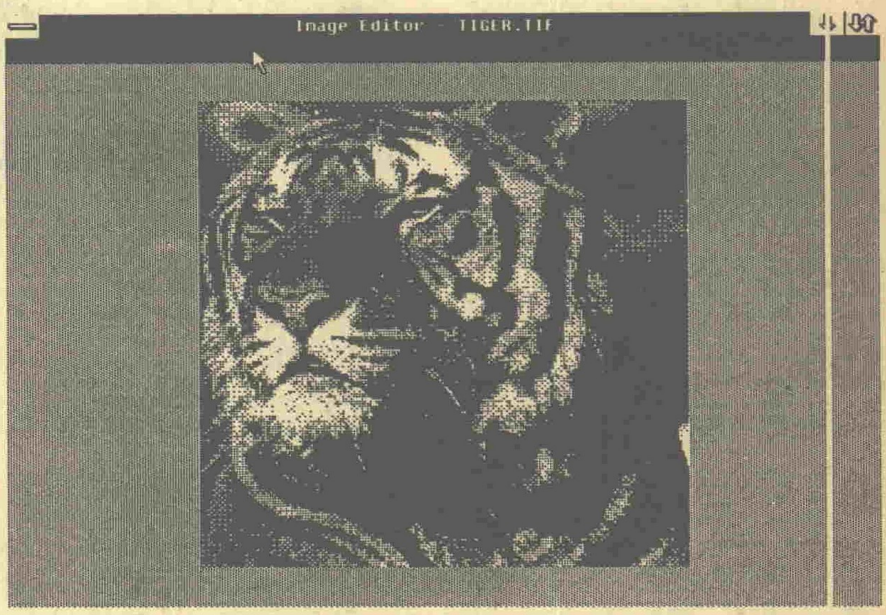
อุปกรณ์ที่นำภาพเข้าส่วนมากก็จะเป็นพวกสแกนเนอร์หรือไม่ก็ Image Grabber

วิธีการแรกนั้นมีลักษณะการทำงานเหมือนเครื่องถ่ายเอกสารคือ แสงจะส่งไปกระทบกับภาพที่ต้องการจะนำเข้าเพื่อให้สะท้อนกับมายังแถวตัวรับแสงซึ่งแบบ CCD หรือ Charge Couple Device ความเข้มของแสงที่สะท้อนกลับมาก็จะถูกแปลงเป็นสัญญาณทางไฟฟ้า นำเข้าไปเก็บคอมพิวเตอร์ต่อไป

ส่วน Image Grabber ก็เป็นพวกที่ดึงเอาสัญญาณภาพจากภายนอกไม่ว่าจะมาจากกล้องหรือจากที่อื่นมาแปลงเป็นสัญญาณดิจิตอลเอาไปเก็บในหน่วยความจำ ซึ่งอาจจะเป็นได้ทั้งสีและขาวดำ

ถ้าจะเปรียบเทียบวิธีการนำเข้าทั้งสองวิธีนี้ จะพบว่าสแกนเนอร์ที่มีขายในท้องตลาดขณะนี้จะเป็นประเภทที่ให้ภาพขาวดำเท่านั้น ส่วนพวกที่ดึงมาจากสัญญาณโดยตรงจะได้ทั้งสีและขาวดำ แต่สแกนเนอร์จะได้ภาพที่มีความละเอียดกว่ามาก ๆ ลองคิดดูง่าย ๆ ยังงี้ สัญญาณภาพที่ใช้ในระบบทีวีนั้นมีความกว้างของแถบความถี่ไม่เกิน 4.5 MHz ในขณะที่สัญญาณภาพที่ได้จาก CGA ธรรมดานั้นสูงถึง 12 MHz

แต่เดิมนั้นสแกนเนอร์ออกแบบมาเพื่อใช้งาน OCR (Opto Character Recognition) เป็นอุปกรณ์รู้จำตัวอักษรสำหรับให้คอมพิวเตอร์อ่านหนังสือเข้าเครื่อง



รูปที่ 6 ภาพขนาด 5" × 6" สแกนโดย HP scan Jet ด้วยความละเอียด 300 จุดต่อนิ้ว ขนาดของแฟ้ม 1 เมกะไบต์ ภาพที่ปรากฏบนจอมีความละเอียดน้อยกว่าที่เก็บไว้ในแฟ้มมาก

โดยตรง

มีสแกนเนอร์ในท้องตลาดขณะนี้มากกว่า 150 ชนิด ตั้งแต่การใช้มือถือลากไปบนรูปภาพ จนถึงชนิดตั้งบนโต๊ะทำงานเหมือนเครื่องถ่ายเอกสาร ราคาตั้งแต่ 200 กว่าเหรียญไปจนถึง 3,000 เหรียญ คุณภาพที่แตกต่างกันก็คือ

- ความละเอียดของการอ่านจากภาพซึ่งบอกเป็นจำนวนจุดต่อนิ้ว

- ความสามารถในการอ่านระดับความเข้มของภาพ

- ความสะดวกสบายในการใช้งาน

ความละเอียดของการอ่านจากภาพมีตั้งแต่ 75 จุดต่อนิ้ว ไปจนถึง 400 จุดต่อนิ้ว ทั้งแบบมือถือและแบบตั้งโต๊ะ

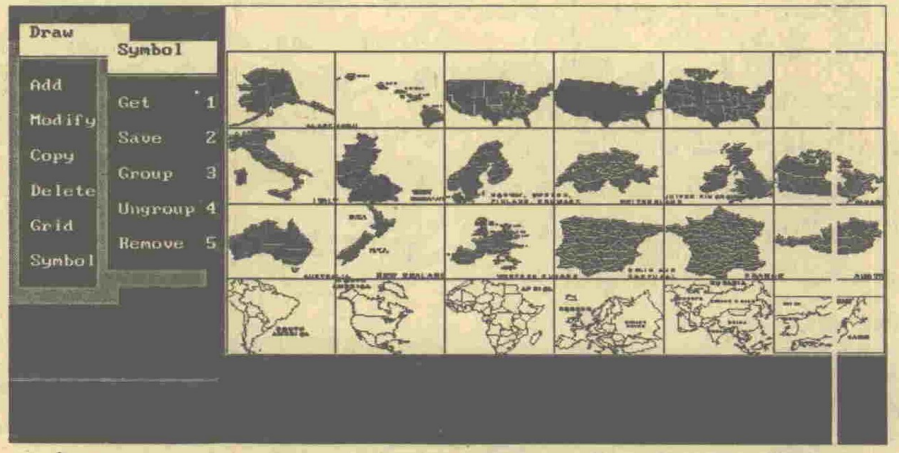
ข้อสำคัญประการหนึ่งก็คือ ความสามารถในการอ่านระดับความเข้มของภาพ เครื่องประเภทมือถือส่วนมากจะอ่านได้ระดับเดียว ส่วนประเภทตั้งโต๊ะจะสามารถอ่านได้ตั้งแต่ 4 ระดับ จนถึง 256 ระดับ

ปัญหาก็คือ ถ้าหากอ่านระดับความเข้มได้แค่ขาวกับดำ ภาพที่อ่านเข้ามาก็หมดรายละเอียดไปมาก เพื่อให้สามารถแสดงระดับความเข้มของสัญญาณได้ เขาก็ใช้เทคนิคที่ใช้กับการพิมพ์รูปในหนังสือพิมพ์มาใช้ ซึ่งเรียก Dither คือใช้ความหนาแน่นของจุดสีดำแทนระดับความเข้มของภาพ

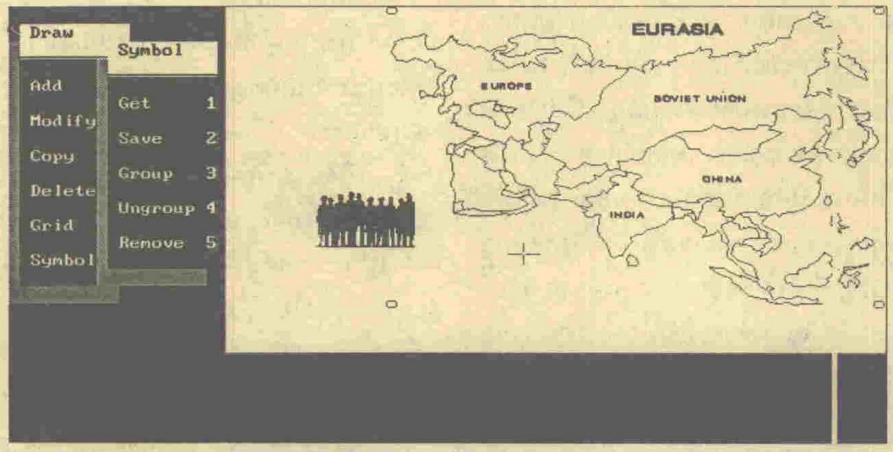
ภาพที่อ่านได้จากสแกนเนอร์พวกนี้จะจัดการใส่ขบวนการ Dither มาให้เสร็จ ส่วนมากภาพที่ได้จากสแกนเนอร์พวกนี้ เขาจะให้มีนามสกุล .PCX (สำหรับ Z-Soft, PC PaintBrush) .PIC สำหรับ Dr.halo แฟ้มพวกนี้สามารถนำมาแก้ไขในลักษณะของบิพ-แม็พโดยตรง

สำหรับสแกนเนอร์ที่สามารถอ่านได้มากกว่า 2 ระดับความเข้ม จะเก็บระดับความเข้มของภาพลงแฟ้มไว้ในลักษณะนั้น เรียกแฟ้มประเภทนี้ว่า TIFF หรือ Tagged Input File Format พอถึงเวลาที่จะต้องนำไปแสดงผล ก็ต้องเปลี่ยนความเข้มของภาพในแต่ละจุดออกมาเป็นความหนาแน่นของจุดแทน (Dither) ดังนั้นแฟ้มพวกนี้จะแก้ไขโดยตรงตามวิธีบิพแม็พไม่ได้ ส่วนมากต้องแปลงเป็นฟอร์แมต์อื่นเสียก่อน

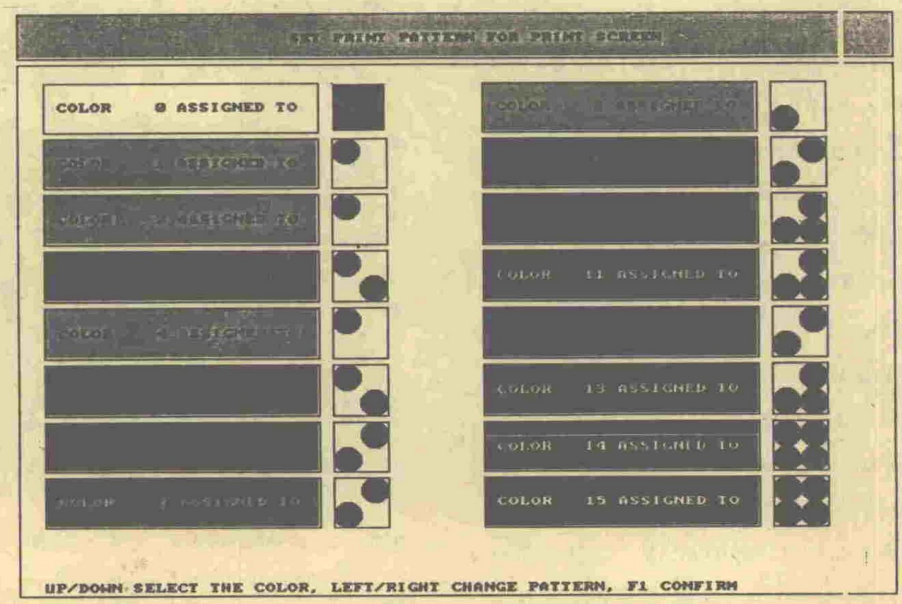
แฟ้มประเภท TIFF จะเป็นแฟ้มที่กินเนื้อที่มาก บางภาพที่มีขนาดเท่ากับ A4 อาจจะต้องใช้ถึง 4 เมกะไบต์ ฉะนั้น



รูปที่ 7 ก. Havard graphic กับตารางสัญลักษณ์



รูปที่ 7 ข. การดึงเอาสัญลักษณ์มาใช้



รูปที่ 8 กำหนดรูปแบบของจุดที่จะพิมพ์สำหรับแทนสีทั้ง 16 สี ของจอ VGA

เครื่องไมโครคอมพิวเตอร์ของท่านอย่างน้อย ๆ ก็ต้องมีฮาร์ดดิสค์เป็นหลักสำหรับการทำงานกับเครื่องสแกนเนอร์

อุปกรณ์ประกอบ

โปรแกรมกราฟิกส่วนมากจะมีส่วนประกอบที่ทำให้การแก้ไขกราฟิกได้ง่ายเข้า นอกเหนือจากการใช้แป้นพิมพ์ในการควบคุมการเขียนแล้ว การใช้เมาส์หรือ Tablet จะทำให้การเขียนกราฟิกทำได้ง่ายเข้า อุปกรณ์พวกนี้บางทีก็เรียกว่า pointing device หรือ Locator device ใน PS/2 จะมีอินพุตสำหรับอุปกรณ์พวกนี้โดยเฉพาะแถมในไบออสยังมีรูทีนสำหรับให้ใช้งานได้อีกต่างหาก

สำหรับเครื่อง PC ธรรมดาทั่ว ๆ ไป pointing device จะต่อเข้าทางช่องการสื่อสารแบบอนุกรม โปรแกรมกราฟิกส่วนมากจะมีไดรเวอร์สำหรับติดต่อกับอุปกรณ์พวกนี้ไว้ให้ การใช้ pointing device ทำให้การใช้งานกราฟิกง่ายกว่ากันเยอะเลย

รูปแบบของแฟ้มกราฟิกชนิดต่าง ๆ

โปรแกรมกราฟิกต่างมีวิธีการเก็บแฟ้มภาพไม่เหมือนกัน แม้ว่าวิธีการที่ใช้ในการสร้างภาพจะมี 2 วิธีใหญ่ ๆ ดังได้กล่าวเอาไว้แล้วในหัวข้อที่ 3 เพื่อให้ท่านได้คุ้นเคยว่าแฟ้มกราฟิกที่มีสกุลแตกต่างกัน มีเชื้อสายมาจากทางไหนบ้าง ก็จะขอเอามารวบรวมเอาไว้เสียตรงนี้

.CGM (Computer Graphic Meta File)

แฟ้มที่อยู่ในสกุลนี้เป็นแฟ้มรูปกราฟิกในลักษณะของเวกเตอร์ และกำลังดำเนินการให้เป็นรูปแบบมาตรฐานสำหรับรูปกราฟิกที่บรรยายวัตถุ (Object-base image) แม้ว่าคณะกรรมการทางด้านมาตรฐานหลายหน่วยยังไม่

รูปที่ 9 โปรแกรม NECPSR.C

```
/*************************************************
 subroutine dump vga graphic screen to printer *
 *************************************************/
  extern char pattern[16];
  char c,sc[12][4],ob[16][3];          /*temporary buffer*/
  int i,j,k,pixl,m,bit,p,col,line;
  /*control string for NEC P2200*/
  char initprn[4]={27,'J',24,0};       /* 24/180 inch line space*/
  char initbmp[6]={27,'*',39,7,5,0};   /*initialize bitmap string*/
  unsigned offset;                     /*180 dot/inch, 1280 dot/line*/

pscr()
 {

  mem2c(&c,0x40,0x49,1);          /*read the video mode*/
  if(c< 4)                        /*not a graphic mode*/
     {ptext();
      return(0);
      }

   /*get the data from screen 12 line at a time  */
    for(line=0;line<40;line++)       /*total 480/12 print line*/
      {
       print(&initbmp[0]);           /*print new line after 1 line*/
       for(i=0;i<21;i++)printc(0);   /*blank first 10 col**/

       for(col=0;col<80;col++)
          {
           /******read data for conversion***/
           for(j=0;j<12;j++)
             for(p=0;p<4;p++)sc[j][p]=r_byte(col,line*12+j,p);

           /***convert to 24 pin of data in stack of 3 byte*/
           for(bit=0;bit<8;bit++)
            {
              for(j=0;j<12;j++)
                 {
                  /****convert pixel value*****/
                  pixl=0;
                  for(p=3;p>=0;p--)
                    {pixl=(pixl<<1)+(sc[j][p]>=128);
                     sc[j][p]=sc[j][p]<<1;
                    }
                  /****read pixel value at x,y**
                   use bios is too slow*********
                  axreg=0xd00;bxreg=0;
                  cxreg=col*8+bit;      /*xlocation*/
                  dxreg=line*12+j;              /*ylocaiot*/
                  vbios();
                  pixl=axreg&0xf;
                  ************/

                  ob[bit*2][j/4]    =((ob[bit*2][j/4])<<2)+(pattern[pixl]&0x3);
                  ob[bit*2+1][j/4]  =((ob[bit*2+1][j/4])<<2)+(pattern[pixl]>>2);
                 }
            } /*next bit*/
           /***now print the 16 for 3 byte dot column**/

           for(i=0;i<16;i++)
             for(j=0;j<3;j++)
                printc(ob[i][j]);

         } /*next screen column**/
        print(&initprn[0]);
        printc(13);
       }    /**next line**/
/*    printc(12);          form feed*/
    return(0);
   }
/************************
```

```
71:    print screen text
72:    ************************/
73:   ptext()
74:    {
75:
76:    k=0;
77:    for(i=0;i<24;i++)
78:     {
79:     for(j=0;j<80;j++)
80:       {
81:         mem2c(&c,0xba00,k,1);
82:         k+=2;
83:         printc(c);
84:         }
85:      printc(13);
86:      printc(10);
87:      }
88:     return(0);
89:    }
90:   print(s)
91:   char *s;
92:    {
93:     while(*s!=0)
94:       {printc(*s);
95:        s++;
96:        }
97:      return(0);
98:     }
```

รูปที่ 9 (ต่อ)

ตกลงได้หมดก็ตาม .CGM ก็ใช้กันมาก ตัวอย่างเช่น Page Maker, Ventura Publisher เป็นต้น

.DRW (Micrografx Draw)

แฟ้มนี้สร้างโดยโปรแกรม Micrografx Draw ซึ่งมีลักษณะเดียวกับแฟ้มของไมโครซอฟต์วินโดว์ที่มีสกุล .PIC สามารถที่จะนำเข้าไปใช้ในเวนทูร่าพับลิชเชอร์ได้

.EPS (Encapsulated Postscript)

แฟ้มพวกนี้ถูกสร้างขึ้นจากการวาดรูปเพื่อนำไปใช้ในโปรแกรมเครื่องพิมพ์บนโต๊ะ (DTP) .EPS เป็นการผสมระหว่างบิตแม็พและเวกเตอร์เข้าด้วยกัน ส่วนที่เป็นเวกเตอร์มาจากภาษาการเขียนกราฟิกที่ชื่อว่า Postcript ในขณะที่ส่วนที่เป็นบิตแม็พอยู่ในรูปของ TIFF (Tagged Input File Format) อุปกรณ์นำออกที่สามารถเข้าใจคำสั่งของ Postcript จึงจะสามารถนำเสนอภาพออกมาได้

.GEM (Graphic Environment Manager)

.GEM เป็นโอเอสในลักษณะของการใช้กราฟิกมาติดต่อกับผู้ใช้ด้วยวิธีเดียวกันกับวินโดว์ ซึ่งเรียกว่า GUI หรือ Graphical user Interfaces รูปแบบของกราฟิกทุกตัวของ .GEM จะเก็บไว้ด้วยสกุล .GEM

.HPGL (Hewlett-Packard Graphics Language)

เป็นรูปแบบที่มีอายุมากที่สุดที่ HP พัฒนาขึ้นมาสำหรับการใช้พล็อตภาพ 2 มิติในพล็อตเตอร์

.IGES (Industrial Graphic Exchange Specification)

งานของ CADD ส่วนมากจะอยู่ในรูปแบบนี้ และใช้โดยหลายบริษัทในการถ่ายโอนข้อมูลทางด้านกราฟิกระหว่างเครื่อง Mini กับ Mainframe หรือระหว่างไมโครคอมพิวเตอร์ด้วยกัน

.PIC (Lotus Picture Files)

เป็นรูปแบบของการบรรยายรูป

รูปที่ 10 โปรแกรม LASERP.C

```
1:  /*************************************************
2:   subroutine dump VGA graphic screen to LASER JET PRINTER *
3:   IN 300 DPI UPFRONT ORIENTED
4:   *************************************************/
5:  extern char pattern[16];
6:  char res[8]={27,'*','t','3','0','0','R',0};     /*resolution 300 DPI*/
7:  char strtp[6]={27,'*','r','0','A',0};           /*start where the cursor is*/
8:  char hdr[9]={27,'*','b','2','3','0','W',0};     /*230 byte of data followed*/
9:  char endg[5]={27,'*','r','B',0};                /*end graphic*/
10: char buffer[4][81];
11:
12: int i,j,k,m,n;
13: char c,c1,c2,c3,c4;
14: char cp1,cp2,pl1[200],pl2[200];
15: pscr()
16:  {
17:
18:   mem2c(&c,0x40,0x49,1);
19:   if(c<4){ptext();bdos(0x4c,0);}
20:
21:   /***initailiaze printer**/
22:   print(&res[0]);
23:   print(&strtp[0]);   /*start to print at thee left margin*/
24:
25:   for(i=0;i<480;i++)              /*start form line 00 to 439*/
26:     {
27:       for(j=0;j<80;j++)
28:         {
29:           c1=r_byte(j,i,0);       /*read plane 0*/
```

รูปที่ 10 (ต่อ)

```
30:           c2=r_byte(j,i,1);       /*read plane 1*/
31:           c3=r_byte(j,i,2);       /*read plane 2*/
32:           c4=r_byte(j,i,3);       /*read plane 3*/
33:
34:           for(n=0;n<2;n++)
35:           {
36:            cp1=0;cp2=0;
37:            for(k=0;k<4;k++)
38:             {
39:               m = (c4)>=128)*8+(c3)>=128)*4+(c2)>=128)*2+(c1)>=128);
40:               cp1=(cp1<<2)+(pattern[m]&03);
41:               cp2=(cp2<<2)+(pattern[m])>>2);
42:
43:               c4=c4<<1;
44:               c3=c3<<1;
45:               c2=c2<<1;
46:               c1=c1<<1;
47:              }
48:             p1[(j)*2+n]=cp1;
49:             p2[(j)*2+n]=cp2;
50:            }  /*for n*/
51:         }    /*for j*/
52:        p1[150]=0;
53:        p2[150]=0;
54:        print(&strtp[0]);              /*start to print at thee left margin*/
55:        print(&hdr[0]);                /*160 byte of data follow*/
56:        for(k=0;k<70;k++)printc(0);    /*insert blank*/
57:        for(k=0;k<160;k++)printc(p1[k]);
58:        print(&strtp[0]);              /*start to print at thee left margin*/
59:        print(&hdr[0]);                /*230 byte of data follow*/
60:        for(k=0;k<70;k++)printc(0);    /*insert blank*/
61:        for(k=0;k<160;k++)printc(p2[k]);
62:        } /***end for i*/
63:      printc(12);
64:      print(&endg[0]);                 /*end graphic mode*/
65:      return(0);
66:      }
67:  /**************************
68:   print screen text
69:   **************************/
70:  ptext()
71:   {
72:
73:   /* printf("Text screen\n");*/
74:   k=0;
75:
76:   for(i=0;i<24;i++)
77:    {
78:    for(j=0;j<80;j++)
79:      {
80:       mem2c(&c,0xb800,k,1);
81:       k+=2;
82:       printc(c);
83:       }
84:     printc(13);
85:     printc(10);
86:     }
87:    printc(12);          /*form feed*/
88:    return(0);
89:   }
90:
91:
92:  print(s)
93:  char *s;
94:    {
95:     while(*s!=0)
96:       {printc(*s);
97:        s++;
98:        }
99:      return(0);
100:     }
```

กราฟิกที่สร้างโดยโลตัสเพื่อการพิมพ์ แม้ว่าจะไม่เหมาะสำหรับกราฟิกที่อยู่ที่หัวแถว กราฟิกรูปแบบนี้ก็ยังยอมรับให้เข้าไปใช้งานได้ใน DTP

.TIF (Tagged Input File Format)

เป็นรูปแบบของการเก็บข้อมูลที่ได้จากการนำภาพเข้า ซึ่งส่วนมากจะมาจากเครื่องสแกนเนอร์ ลักษณะของการเก็บแฟ้มพวกนี้จะเป็นบิตแม็พที่มีรหัสความเข้มของแต่ละพิกเซลรวมอยู่ด้วย

.WMF (Windows Metafile Format)

รูปแบบนี้สร้างขึ้นมาเพื่อการถ่ายโอนแฟ้มระหว่างโปรแกรมที่ใช้ไมโครซอฟต์วินโดว์เป็นโอเอส และโปรแกรมที่ไม่ได้ใช้วินโดว์เป็นฐาน ข้อเสียของแฟ้ม .WMF ก็คือแฟ้มจะใหญ่เกิน 64 กิโลไบต์ ไม่ได้

ของแถมสำหรับท่านที่มี VGA

ก่อนจะจบ ผมมีของแถมมาฝากท่านที่ใช้จอ VGA หรือ EGA คือ โปรแกรม Dump หน้าจอกราฟิกออกที่เครื่องพิมพ์ ดังที่ผมพิมพ์ออกมาเป็นรูปประกอบในบทความนี้ ท่านอาจจะใช้บริการซอฟต์เมล์ของไมโครฯ ขอโปรแกรมเหล่านี้ได้ทั้งแฟ้ม .EXE และซอร์สโค้ด

ปัญหาการ Dump หน้าจอของจอ EGA หรือ VGA ก็คือสีของพิกเซลที่ปรากฏบนจอที่พิมพ์ออกที่เครื่องพิมพ์ควรจะมีความเข้มของการพิมพ์ที่แตกต่างกันเพื่อให้ได้ภาพที่สมจริงกับที่ปรากฏบนจอ ในโปรแกรมนี้จะใช้เทคนิค Dither มาแปลงสีให้เกิดความเข้มของภาพที่ต่างกัน โดย 1 พิกเซลบนหน้าจอจะแทนด้วยจุดพิมพ์ 2 × 2 จุด โดยกำหนดให้แต่ละสีมีรูปแบบการพิมพ์ที่แตกต่างกันดังแสดงในรูปที่ 6

โปรแกรม Dump สกรีนนี้จะเข้าไปแทนอินเตอร์รัพต์เวกเตอร์ 5 ซึ่งทำหน้าที่ดั๊มพ์สกรีน และฝังตัวเองอยู่ในหน่วย

ความจำ โปรแกรมที่เขียนขึ้นมายังเป็นโปรแกรมใช้กับเครื่องพิมพ์เฉพาะตัวอยู่ ในที่นี้ผมทำไว้ 2 ชนิดคือ สำหรับเครื่องพิมพ์ NEC P2200 และ HP Laser Jet Series II ถ้าเป็นเครื่องพิมพ์ดอตเมตริกซ์ 24 จุดพิมพ์เช่นเดียวกับ NEC P2200 ลองใช้ NECPSCR.EXE ดู ถ้าเป็น HP Laser Jet ให้ใช้ LASERP.EXE

หลังจากโหลดโปรแกรมฝังตัวเองในหน่วยความจำแล้วกด Print screen ได้ทันที ถ้าเป็นเท็กซ์โหมดก็จะพิมพ์เท็กซ์ให้ ถ้าเป็นกราฟิก (VGA หรือ EGA 16 สีเท่านั้น) ก็จะพิมพ์กราฟิกให้

ถ้าโปรแกรมกราฟิกที่ไม่ไปเปลี่ยนแปลงอินเตอร์รัพต์ 5 ท่านจะสามารถพิมพ์หน้าจอกราฟิกออกมาได้โดยใช้แป้น Print screen

เมื่อเริ่มต้นโปรแกรม รูปแบบที่ใช้แทนสีจะแตกต่างกันตามลำดับ ดังในรูปที่ 8 ถ้าหากท่านจะเปลี่ยนแปลงรูปแบบที่ใช้แทนสีใหม่ ใช้โปรแกรม SETPSCR.EXE สำหรับเปลี่ยนแปลงรูปแบบ โดยจะให้ท่านเข้าไปเปลี่ยนแปลงรูปแบบ ดังในรูปที่ 7 โดยการใช้แป้นควบคุมเคอร์เซอร์ UP/DOWN เลือกสีที่ต้องการเปลี่ยนรูปแบบใช้แป้นควบคุมเคอร์เซอร์ Left/Right สำหรับเปลี่ยนรูปแบบของสีนั้น ๆ กด F1 เมื่อทุกอย่างเรียบร้อย

หากต้องการจะเปลี่ยนรหัสควบคุมให้ตรงกับเครื่องพิมพ์ที่ท่านมีก็ลองเปลี่ยนซอร์สโค้ดในส่วนที่ควบคุมการพิมพ์ เสร็จจากนั้นแล้วคอมพายล์ออกมาเป็นออปเจคโค้ดนำมาลิงค์เข้าด้วยกัน

```
>Link SCRDUMP + NECPSR,
NECPSCR,,
```

สำหรับเครื่องพิมพ์ดอตแมตริกซ์ 24 จุดพิมพ์

```
>Link SCRDUMP + LASERP,
LASERP,,,
```

สำหรับ Laser Jet

ลองดูแล้วใช้ได้ผลประการใดบอกกันให้รู้บ้าง

รูปที่ 11 โปรแกรม SCRDMP. ASM

```
1:  ;PRINT GRAPHIC SCREEN
2:  ;TO BE USE WITH PSCR.C
3:  ;PRINT ON COMMAND PRINT SCREEN
4:  ;WRITTEN BY DR.PAISAN   DEC 89
5:
6:  DOS_FUN    EQU 21H          ;DOS FUNCTION CALL
7:  DOS_RES    EQU 27H
8:  GET_VEC    EQU 35H
9:  SET_VEC    EQU 25H
10:
11: DGROUP GROUP DATA
12: DATA  SEGMENT PUBLIC 'DATA'
13:         ;***DEFAULT BLACK & WHITE PATTERN***
14:         PUBLIC PATTERN
15: PATTERN DB 0            ;COLOR 0
16:         DB 01H          ;COLOR 1
17:         DB 02H          ;COLOR 2
18:         DB 03H          ;COLOR 3
19:         DB 04H          ;COLOR 4
20:         DB 05H          ;COLOR 5
21:         DB 06H          ;COLOR 6
22:         DB 07H          ;COLOR 7
23:         DB 08H          ;COLOR 8
24:         DB 09H          ;COLOR 9
25:         DB 0AH          ;COLOR 10
26:         DB 0BH          ;COLOR 11
27:         DB 0CH          ;COLOR 12
28:         DB 0DH          ;COLOR 13
29:         DB 0EH          ;COLOR 14
30:         DB 0FH          ;COLOR 15
31:
32: MAS62 DB 'PRINT GRAPHIC SCREEN ON REQUEST IS ALREADY INSTALLED  ',13,10,'$'
33: MASG  DB 'PRINT GRAPHIC SCREEN IS INSTALLED ',13,10
34:       DB 'USE PrintScreen TO PRINT GRAPHIC  ',13,10,'$'
35: OLD_V DD  0             ;SAVE OLD KEY VECTOR HERE
36: DATA  ENDS
37:
38: PGROUP GROUP PROG
39: PROG  SEGMENT PUBLIC 'PROG'
40:       ASSUME CS:PROG,DS:PROG
41:       EXTRN  PSCR:NEAR
42:       PUBLIC  PRINTC,INIT,R_BYTE,MEM2C
43: MAIN:  JMP  INIT
44:
45:
46: DSEG        DW   DATA
47:             DB  '*'
48: MARKER      DB  '@'         ;MARKER ID
49:      ;PRINT TIME WHILE WAITING FOR KEY
50: PRINTSCR PROC NEAR
51:      ;SAVE THE REGISTER USED HERE
52:      STI
53:      PUSH  BP
54:      PUSH  DI
55:      PUSH  SI
56:      PUSH  CX
57:      PUSH  AX
58:      PUSH  BX
59:      PUSH  DX
60:      PUSH  DS
61:      PUSH  ES
62:      MOV   AX,CS
63:      MOV   DS,AX
64:      MOV   AX,DSEG
65:      MOV   DS,AX            ;GET THE DATA SEGMENT
66:      CALL  PSCR
67: XTIME:
68:      POP  ES
69:      POP  DS
70:      POP  DX
```

```
71:       POP  BX
72:       POP  AX
73:       POP  CX
74:       POP  SI
75:       POP  DI
76:       POP  BP
77:       IRET
78:  PRINTSCR ENDP
79:  ;*****************************
80:  ;PROCEDURE PRINT CHARACTER TO PRINTER
81:  PRINTC PROC NEAR
82:       PUSH  BP
83:       MOV   BP,SP
84:       MOV   DX,0          ;PRINTER NUMBER 0
85:       ;TEST PRINTER
86:  PTEST:
87:       MOV   AX,0200H;
88:       INT   17H
89:       AND   AH,81H
90:       JZ    PTEST
91:       MOV   AX,[BP+4]       ;CHARACTER TO PRINT
92:       MOV   AH,0
93:       INT   17H
94:       POP   BP
95:       RET
96:  PRINTC  ENDP
97:  ;*************************************
98:  ;READ 1 BYTE OF VIDEO RAM
99:  ;*************************************
100: R_BYTE PROC NEAR
101:       PUSH  BP
102:       MOV   BP,SP
103:
104:       MOV   BX,[BP+4]    ;BYTE NUMBER
105:       MOV   AX,[BP+6]    ;LINE NUMBER
106:       MOV   DX,80
107:       MUL   DX
108:       ADD   AX,BX        ;ADDRESS IN THE VRAM
109:       MOV   SI,AX
110:       MOV   AX,[BP+8]    ;PLANE NUMBER
111:       MOV   AH,AL
112:       MOV   AL,4         ;READ PLANE REGISTER
113:       MOV   DX,03CEH
114:       OUT   DX,AX        ;SELECT THE READ PLANE
115:       PUSH  ES
116:       MOV   AX,0A000H    ;VIDEO SEGMENT
117:       MOV   ES,AX
118:       MOV   AX,ES:[SI]
119:       POP   ES
120:       POP   BP
121:       RET
122: R_BYTE  ENDP
123:
124:       assume cs:PROG, ds:DATA
125:
126: INIT PROC NEAR
127:       MOV  AX,DATA
128:       MOV  DS,AX
129:       ;SAVE OLD KEY BOARD INT
130:       MOV  AL,5              ;get vector of the printscreen routine
131:       MOV  AH,GET_VEC
132:       INT  DOS_FUN
133:
134:       ;TEST FOR REPETITION OF THE INSTALL
135:       MOV  AL,ES:[BX-1]
136:       CMP  AL,'#'
137:       JNZ   NREADY      ;IF '#' IS MARKED BEFORE THE INT. THEN
138:                         ;ROUTINE IS ALREADY INSTALLED
139:
140:       MOV  DX,OFFSET MAS62
141:       MOV  AH,09H
142:       INT  21H
143:       JZ   INITRET
144:
145:  NREADY:
146:       MOV  WORD PTR OLD_V,BX
147:       MOV  WORD PTR OLD_V[2],ES  ;SAVE OLD KEY IO IN THE MEMORY
148:
149:       PUSH DS
150:       MOV  AX,CS
151:       MOV  DS,AX
152:       MOV  DX,OFFSET PRINTSCR
153:       MOV  AL,5             ;REPLACE PRINTSCREEN INTERRUP
154:       MOV  AH,SET_VEC       ;USE DOS FUNCTION TO DO THIS
155:       INT  DOS_FUN          ;CALL DOS FUNTION
156:       POP  DS
157:       MOV  DX,OFFSET MAS6
158:       MOV  AH,09H
159:       INT  21H
160:
161:       MOV  DX,OFFSET INIT
162:       ADD  DX,10000
163:       ;EVERY THING IS SET AND READY TO GO
164:       INT 27H             ;TERMINATE BUT STAY RESIDENT
165:  INITRET:
166:       MOV  AX,4C00H
167:       INT 21H          ;PROGRAM TERMINATE
168:  INIT  ENDP
169:
170:  ;TRANSFER DATA TO C MEMORY BUFFER
171:  ;SYNOPIS  MTRANSF(B,SEG,OFS,LEN)
172:  ;          CHAR *B        /*BUFFER TO TRANSFER TO
173:  ;          INT   SEG      /*SEGMENT OF TO SOURCE
174:  ;          INT   OFS      /*OFFSET OF THE SOURCE
175:  ;          INT   LEN      /*LENGTH OF DATA TO BE TRANSFERED
176:  ;*****************************************
177:            ASSUME CS:PGROUP
178:  MEM2C     PROC   NEAR
179:            PUSH   BP
180:            MOV    BP,SP
181:            PUSH   DS
182:            PUSH   ES
183:
184:            MOV    AX,DS
185:            MOV    ES,AX
186:            MOV    DI,[BP+4]      ;C BUFFER
187:            MOV    AX,[BP+6]      ;SEGMENT OF THE SOURCE
188:            MOV    DS,AX
189:            MOV    AX,[BP+8]      ;OFFSET OF THE SOURCE
190:            MOV    SI,AX
191:            MOV    CX,[BP+10]     ;LENGHT OF DATA
192:            CLD
193:            REP    MOVSB
194:            POP    ES
195:            POP    DS
196:            POP    BP
197:            RET
198:  MEM2C     ENDP
199:  PROG ENDS
200:       END MAIN
201:
202:
203:
204:
205:
206:
207:
208:
209:
210:
211:
212:
```